Eight Complete Programs in this Issue

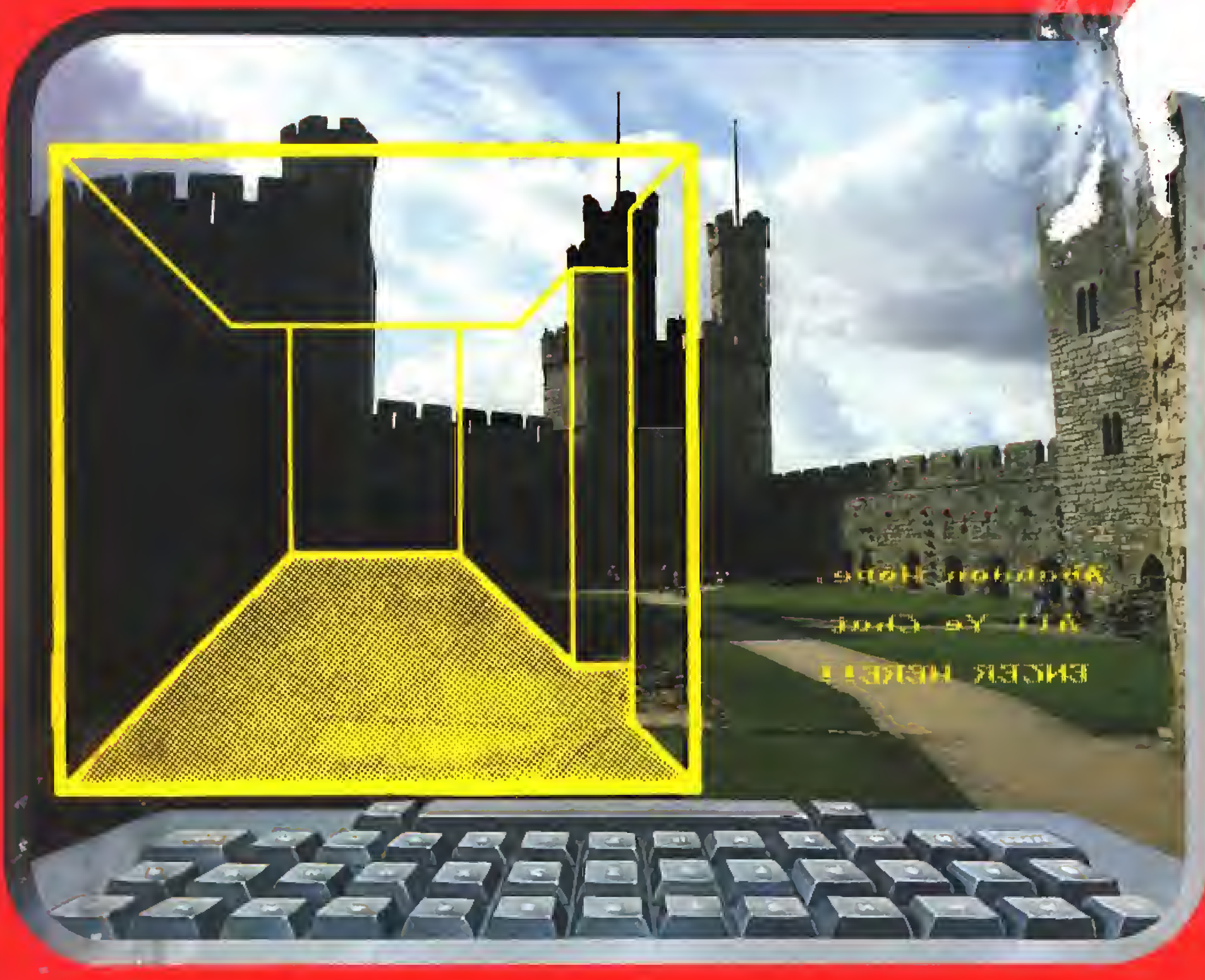
NO. 74

U.S. Edition: $2.50
International Edition: $3.50

AUGUST 1984

MICRO™

for the Serious Computerist



Useful Math Functions
3-D Mazes
Step/Trace Programs
HiRes Graphic Printouts
p-System Comparisons

0 74470 11690


MORE THAN JUST THE TIP
subscribe to!
The Transactor
The Tech/News Journal For Commodore Computers Vol. 4
...AND GET MORE FROM YOUR COMMODORE 64 AND VIC 20
"The Transactor is ... constant resource." JIM BUTTERFIELD
"The Material is ... Very impressive. ... friendly tone of the Transactor." -LIZ DEAL
CHECK THESE FEATURES:
• Up-to-date: new ... new products and services.
• Insights and stra... talk from respected exper... Jim Butterfield.
• Interesting bits o... among Commodo... users.
SIMPLY THE BEST
Subscribe now a... you'll wonder how yo... managed without it.
THE TRANSACTOR SUBSC... DEPT.
500 Steeles Avenue, Milton, O... L9T 3P7
Telephone: 876-4741
Year's subscription (6 issues) $1...
Back issues available.

editorial

Sharing Information

I recently received a letter from a reader enquiring whether he could use one of our programs in a software package he was developing. As in any healthy field, a fair amount of cross-fertilization takes place within. This is particularly true in the world of software development where one great idea will spawn another.

Now in the midst of all of this comes that ancient cry of plagarism. It is said that "mimicry is the sincerest form of flattery" but not all authors enjoy such attention. Where do you draw the line between what is plagarism and what is a logical or original development of an idea? It seems more and more that this is a fine line — one that is made by the creator rather than the lawyer. Times are changing. There was a time when everyone 'borrowed' ideas from everyone else. Centuries ago in the music world this was common practice. But attitudes changed over the years to the point that people horded their ideas as their own and sued anyone who dared approach any similarity to their mental offspring. Now once again the tides are shifting in the other direction. This shift is putting many wonderful and original ideas in our hands. It is furthering new creations and saving a lot of wasted time in the process.

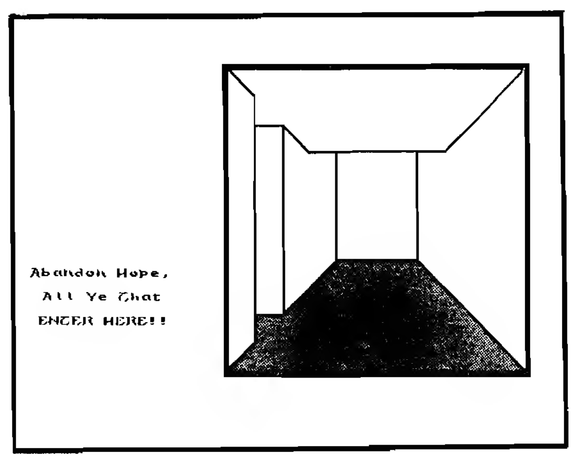
How is this happening? To start with, for the first time software producers are providing software that is not copy protected. This allows for backup and working copies to be easily made. Now you can have multiple copies of handy programs on many disks, freeing you up from playing musical disks. Beyond this there has been an even bolder step taken by some publishers. They state you can utilize their product in your own software without any penalty and at nominal cost. The usual agreement is that you simply register yourself and the product with the publisher and give acknowledgement of the use of the program. Penguin Software has done this with a number of their graphics packages. In particular I cite their Graphics Magician Picture Painter which is not copy protected and allows for its use in other products. Indeed, many people have taken advantage of the offer. In permitting use of the Graphics Magician, Penguin has saved people hours of laboriously drawing graphics, since it allows you to create and save hundreds of pictures on a single disk where normally only 11-15 pictures would fit. Here is a perfect example of how this willingness to share ideas frees up an author to do what he otherwise might not be able to do because of time expenditure or lack of expertise. He suddenly can add professional graphics to a software package that he is working on without any difficulty. He is not forced to learn what may be a completely foreign area of programming, but can instead benefit from someone elses expertise and experience.

MICRO feels that its place is one of a disseminator of knowledge, a place where people come to learn new things and share what they have learned. In this spirit we encourage people to **incorporate** what they discover in MICRO in their own work. The word **incorporate** is very significant in this context. If you are just using the material for your own personal benefit, then that is your right. That's why you buy MICRO in the first place. However, MICRO can not allow its **major programs** to be copied and distributed by others — be they individuals, companies, computer clubs, etc. This would lessen the marketability of MICRO Magazine, the MicroDisks, books and software packages. The **smaller programs** and **utility routines** are a different matter. If you have a project that can benefit by **incorporating** some MICRO material, please contact us. Normally we will grant you permission to incorporate the material in your product, subject to signing a license that protects all parties and agreeing to a proper acknowledgement. In this way we feel we serve you better and make a greater pool of ideas available to more people. Perhaps a program that we publish will find its way into the hands of someone who will turn it into something we can use. And then, in turn, we may be inspired to create something else which will start the circle again. The end result is a continued growth personally and professionally for all involved - you and MICRO.

Mark S. Morano

Mark S. Morano
Technical Editor

On The Cover



The tortuous passageways of Carnaervon Castle in Wales are the scene of our 3-D Maze. The usual warning is, of course, given to all foolish enough to enter, printed with the Atari/Epson custom characters offered by Mike Bassman's program.

Photo by Cindy Kocher

highlights

Featured This Month

This month we offer eight complete major programs on a variety of topics, plus an excellent look at the 68000 machines and their operating systems.

Truly 3-D Mazes — generate truly 3-D mazes with a minimum of effort and memory. You get a rat's-eye view of the maze corridors as you "walk" through! And it all starts with a cube and 3 1/2K of RAM!

Alter Track and Sector — allows the user to dump, in hex, any sector on a disk with VIC/Commodore format, then modify any byte in that sector. Rewrite lost headers without loss of data!

Extended Precision Arithmetic — if greater mathematical precision is needed than is usual in BASIC, for statistical calculations perhaps, then this is the program for you. An interesting application is included in what may be the only program for calculating the lunar-based Jewish Calendar.

Relocatable Step/Trace — this step and trace routine can be easily moved to any part of a program.

The UCSD p-System — a careful, lucid explanation of how the fast, flexible p-system works and why it is becoming THE 68000 Operating System. Also, a review of six 68000-based microcomputers which puts the new systems in perspective.

Atari Character Printer — creating an unending array of different character fonts on your Atari screen can be fun, but now you can also print them out on paper with complete accuracy!

Useful Math Functions — save time and mathematical aggrevation with a collection of defined functions.

C-64 Graphic Dump — this "perfect" dump works either in HiRes or multi-color, allows large size printouts and is very fast. This month, learn to interface 5 major commercial packages - whatever you generate you can now dump.

HILISTER — highlighting lines of text and programs can be useful for emphasis or clarity when discussing material on the screen in business meetings, classrooms, seminars. This program also allows easy movement within a program or text.

MICRO™

MICRO is published monthly by:
MICRO, Chelmsford, MA 01824.
Second Class postage paid at:
Chelmsford, MA 01824 and additional mailing offices.
USPS Publication Number: 483470.
ISSN: 0271-9002.
Send subscriptions, change of address, USPS Form 3579, requests for back issues and all othe fulfillment questions to:
MICRO
P.O.Box 6502
Chelmsford, MA 01824
or call 617/256-3649.
Subscription Rates: (per year):
U.S. $24.00 or $42.00 for two years
Foreign surface mail: $27.00
Air mail: Europe $42.00
Mexico, Central America, Middle East, North Africa, Central Africa $48.00
South America, South Africa, Far East, Australia, New Zealand $72.00

MICRO™

for the Serious Computerist

AUGUST 1984

NO.74

Product Reviews

Departments

It all adds up...

HOME COMPUTERS

ATARI®

600XL $149
800XL $229
WHILE SUPPLY LASTS

1200XL .... CALL
1450XL .... CALL

850 Interface	$159.00
1010 Recorder	$71.99
1020 Color Printer	$219.00
1025 Dot Matrix Printer	$299.00
1027 Letter Quality Printer	$269.00
1030 Direct Connect Modem	$99.99
1050 Disk Drive	$349.00
1064 Memory Module	$125.00
Touch Tablet/Software	$64.99
Light Pen/Software	$72.99
CX22 Track Ball	$39.99
7097 Atari Logo	$74.99
4018 Pilot (Home)	$57.99
405 Pilot (Educ.)	$99.99
8036 Atari Pilot	$77.99
5049 VisiCalc	$149.99
488 Communicator II	$119.99
CX30 Paddles	$11.99
CX40 Joystick	$7.99
4011 Star Raiders	$31.99
4022 Pac Man	$31.99
4025 Defender	$31.99
8026 Dig Dug	$31.99
8031 Donkey Kong	$35.99
8034 Pole Position	$37.99
8040 Donkey Kong Jr	$33.99
8043 Ms Pacman	$37.99
8044 Joust	$37.99
8045 Pengo	$33.99
8052 Moon Patrol	$33.99
4003 Assembler	$44.99
8126 Microsoft Basic I or II	$64.99

DISK DRIVES FOR ATARI

PERCOM
AT 88-S1 ... $279.00
AT 88-S1 PD ... $299.00

INDUS
GT Drive ... $379.00

RANA
1000 ... $299.00

TRAK
AT-D2 ... $389.00
AT-D4 ... $589.00

MEMORY BOARDS

ATARI
Axlon 32K ... $59.99
Axlon 48K ... $99.99
Axlon 128K ... $299.00

APPLE/FRANKLIN
Axlon 128K ... $299.00
Axlon 320K ... $849.00

ALIEN VOICE BOX

Atari ... $119.00
Apple ... $149.00

DISKETTES

MAXELL
5¼" MD-1 ... $24.99
5¼" MD-2 ... $34.99
8" FD-1 ... $39.99
8" FD-2 ... $49.99

VERBATIM
5¼" SS/DD ... Value Life ... $22.99
5¼" DS/DD ... Value Life ... $29.99

BIB
5¼" Disk Head Cleaner ... $14.99

ELEPHANT
5¼" SS/SD ... $18.49
5¼" SS/DD ... $21.99
5¼" DS/DD ... $26.99

DISK HOLDERS
INNOVATIVE CONCEPTS
Flip-n-File 10 ... $3.99
Flip-n-File 50 ... $17.99
Flip-n-File 50 w/lock ... $24.99
Flip-n-File (400/800 ROM) ... $17.99

CONTROLLERS & JOYSTICKS

WICO
Joystick ... $21.99
3-way Joystick ... $22.99
Power Grip ... $21.99
BOSS Joystick ... $17.99
ATARI/VIC Trak Ball ... $34.99
Apple Trak Ball ... $54.99
Apple Analog ... $37.99

KRAFT
Joystick ... $41.99
Atari Single Fire ... $12.99
Atari Switch Hitter ... $15.99
Apple Paddles ... $34.99
IBM Paddles ... $34.99
IBM Joystick ... $46.99

Koala
Atari (ROM) ... $79.99
C-64 (ROM) ... $79.99
IBM ... $99.99
Apple/Franklin ... $85.99

PRINTERS

AXIOM
AT-100 Atari Interface Printer ... $239.00
GP-100 Parallel Interface ... $199.00
GP-550 Atari Bidirectional ... $319.00
GP-700 Atari Color Printer ... $489.00
GP-550 Parallel Printer ... $269.00

BMC
401 Letter Quality ... $589.00
BX-80 Dot Matrix ... $269.00

C.ITOH
Gorilla Banana ... $209.00
Prowriter 8510P ... $379.00
Prowriter 1550P ... $599.00
A10 (18 cps) ... $569.00
Hot Dot Matrix ... CALL
F10-40 ... $999.00
F10-55 ... $1349.00

COMREX
ComWriter II Letter Quality ... $499.00

DIABLO
620 Letter Quality ... $949.00
630 Letter Quality ... $1749.00

DAISYWRITER
2000 ... $999.00
Tractor Feed ... $109.00

EPSON
RX-80, RX-80FT, RX-100 ... CALL
FX-80, FX-100 ... CALL
LQ 1500 ... CALL

IDS
Prism 80...For Configuration ... CALL
Prism 32...For Configuration ... CALL

JUKI
6100 ... $499.00

MANNESMAN TALLY
160L ... $589.00
180L ... $799.00
Spirit 80 ... $309.00

NEC
8023 Dot Matrix ... $389.00
8025 Dot Matrix ... $669.00
2010/15/30 ... $749.00
3510/15/30 ... $1369.00
7710/15/30 ... $1799.00

OKIDATA
82, 83, 84, 92, 93, 2350, 2410 ... CALL

OLYMPIA
Compact 2 ... $479.00
Compact RO ... $509.00
ESW 3000 ... $1449.00

SMITH CORONA
TP-1000 ... $449.00
Tractor Feed ... $119.00

SILVER REED
500 Letter Quality ... $449.00
550 Letter Quality ... $549.00
770 Letter Quality ... $899.00

STAR
Gemini 10X ... $299.00
Gemini 15X ... $399.00
Serial Board ... $75.00
Radix 10 ... $599.00
Radix 15 ... $699.00

TOSHIBA
1340 ... $889.00
1351 ... $1699.00

TRANSTAR
120P ... $489.00
130P ... $649.00
315 Color ... $459.00

MODEMS

ANCHOR
Volksmodem ... $59.99
Mark IL Serial ... $79.99
Mark VII (Auto Ans/Auto Dial) ... $99.99
Mark XII (1200 Baud) ... $299.99
Mark TRS-80 ... $99.99
9 Volt Power Supply ... $9.99

HAYES
Smartmodem 300 ... $209.00
Smartmodem 1200 ... $499.00
Smartmodem 1200B ... $449.00
Micromodem IIe ... $269.00
Micromodem 100 ... $299.00
Smart Com II ... $89.99
Chronograph ... $199.00

NOVATION
J-Cat ... $99.99
Cat ... $139.00
Smart Cat 103 ... $179.00
Smart Cat 103/212 ... $399.00
AutoCat ... $219.00
212 AutoCat ... $549.00
Apple Cat II ... $249.00
212 Apple Cat ... $449.00
Apple Cat 212 Upgrade ... $259.00
PC Cat Access 1 2 3 ... $399.00

ZENITH
ZT-1 ... $339.00
ZT-10 ... $309.00
ZT-11 ... $369.00

MONITORS

AMDEK
300 Green ... $149.00
300 Amber ... $159.00
310 Amber ... $169.00
Color 1 ... $279.00
Color 1 Plus ... $299.00
Color 2 Plus ... $419.00
Color 3 ... $349.00
Color 4T IBM ... $699.00

BMC
1201 (12" Green) ... $88.99
1201 Plus (12" Green Hi-Res) ... $98.99
9191 Plus ... $249.00

GORILLA
12" Green ... $88.99
12" Amber ... $95.99

NEC
JB 1260 Green ... $109.00
JB 1201 Green ... $149.99
JB 1205 Amber ... $159.99
JB 1215 Color ... $259.00
JC 1216 RGB ... $429.00
JC 1460 Color ... $359.00

PRINCETON GRAPHICS
MAX 12 Amber ... $199.00
HX-12 RGB ... $539.00
SRT-2 RGB ... $649.00

SAKATA
SC 100 Color ... $269.00
SG-1000 Green ... $129.00
SA-1000 Amber ... $139.00

TAXAN
210 Color RGB ... $299.00
400 Med-Res RGB ... $319.00
415 Hi-Res RGB ... $439.00
420 Hi-Res RGB (IBM) ... $489.00
100 12" Green ... $125.00
105 12" Amber ... $135.00

USI
Pi 1 9" Green ... $99.99
Pi 2, 12" Green ... $119.99
Pi 3, 12" Amber ... $129.99
Pi 4, 9" Amber ... $119.99
1400 Color ... $269.99

QUADRAM
Quadchrome 8400 Color ... $519.00

ZENITH
ZVM122 Amber ... ~~$99.99~~
ZVM123 Green ... $89.99
ZVM124 IBM-Amber ... ~~$149.00~~
ZVM131 Color ... $309.00
ZVM133 RGB ... $429.00
ZVM135 RGB Composite ... $469.00

COMPUTER MAIL ORDER

west
VISA
800-648-3311
In NV call (702)588-5654, Dept. 125
Order Status Number: 588-5654
P.O.Box 6689
Stateline, NV 89449

canada
Ontario/Quebec 800-268-3974
Other Provinces 800-268-4559
In Toronto call
(416)828-0866, Dept. 125
Order Status Number: 828-0866
2505 Dunwin Drive, Unit 3B
Mississauga, Ontario, Canada L5L1T1

east
MasterCard
800-233-8950
In PA call (717)327-9575, Dept. 125
Order Status Number: 327-9576
Customer Service Number: 327-1450
477 E. 3rd St., Williamsport, PA 17701

No risk, no deposit on C.O.D. orders and no waiting period for certified checks or money orders. Add 3% (minimum $5) shipping and handling on all orders. Larger shipments may require additional charges. NV and PA residents add sales tax. All items subject to availability and price change. Call today for our catalog.

the best prices

APPLE/FRANKLIN DISK DRIVES

MICRO-SCI

A2 ... $219 00
A40 ... $299 00
A70 ... $319 00
C2 Controller ... $79 99
C47 Controller ... $89 99

RANA

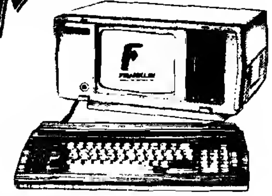
Elite 1 ... $279 00
Elite 2 ... $389.00
Elite 3 ... $589.00

APPLE IIe STARTER PACK

64K Apple IIe, Disk Drive & Controller, 80 Column Card, Monitor II & DOS 3.3
COMPLETE ... CALL

Call on all other Apple Models

FRANKLIN



ACE 1000 Color Computer ... CALL
ACE PRO PLUS System ... CALL
ACE 1200 Office Mgmt. System ... CALL
ACE PORTABLES ... CALL

NEC PRINTERS

NEC 2050 ... $899.00
NEC 3550 ... $1869.00

PERCOM/TANDON DISK DRIVES

5¼" 320K Floppy ... $219.00
5 Meg Hard w Controller ... $1049 00
10 Meg Hard w Controller ... $1349 00
20 Meg Hard w Controller ... $1899 00

VISICORP

VisiCalc IV ... $159.00
VisiWord + ... $249.00
Visi-on Application Manager ... $79.99
Visi-on Calc ... $289.00
Visi-on Graph ... $179.00
Visi on Word ... $249.99
Optical Mouse ... $189.99

AST RESEARCH

Six Pak Plus ... from ... $279 00
Combo Plus II ... from ... $279.00
Mega Plus ... from ... $309.00
I/O Plus ... from ... $139.00

QUADRAM

Quadlink ... $479.00
Quadboard ... as low as ... $289 00
Quad 512 Plus ... as low as ... $249 00
Quadcolor ... as low as ... $219.00
Chronograph ... $89 99
Parallel Interface Board ... $89.99
64K RAM Chips Kit ... $59.99

IBM

MICROPRO

WordStar Professional Pack ... $389 00
InfoStar ... $299 00
SpellStar ... $159 00
CalcStar ... $99 99

MICROSTUF

Crosstalk ... $105.00

MICROSOFT

MultiPlan ... $159.00

ASHTON-TATE

dBASE II ... $339 00
dBASE III ... $449 00
Friday! ... $185 00

IUS

EasyWriter II ... $249 00
EasySpeller ... $119.00
EasyFiler ... $229.00

CONTINENTAL SOFTWARE

1st Class Mail/Form Letter ... $79 99
Home Accountant Plus ... $88.99

LOTUS

Symphony ... $549 00
1-2-3 ... $339 00

PROFESSIONAL SOFTWARE

PC Plus/The Boss ... $329 00

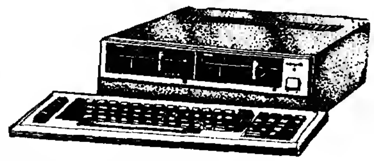
SYNAPSE

File Manager ... $89.99

MBC 550 ... CALL
MBC 555 ... CALL
MBC 555-2 ... CALL

MBC 1100 ... $1499.00
MBC 1150 ... $1899.00
MBC 1200 ... $1849 00
MBC 1250 ... $2099 00
PR 5500 Printer ... $599 00

SANYO

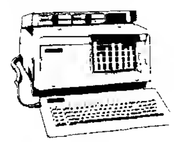


HEWLETT PACKARD

HP 71B ... $419.99
41CV ... $189.99
41CX ... $249.99

HP 11C ... $62.99
HP 12C ... $92.99
HP 15C ... $92.99
HP 16C ... $92 99
HP 75D ... $879 99
HPIL Module ... $98 99
HPIL Cassette or Printer ... $359.99
Card Reader ... $143 99
Extended Function Module ... $63 99
Time Module ... $63 99

ZENITH data systems



PC COMPATIBLE 16 BIT SYSTEMS

Z 150 PC — Z 160 PC

Call for price and configurations

SOFTWARE

	C-64	Atari	IBM	Apple
Electronic Arts				
One on One	$29 99	$29.99	$29.99	$29.99
Music Construction	$29.99	$29.99	$29.99	$29.99
Pinball Construction	$29.99	$29 99	$29 99	$29.99
Cut & Paste	$39.99	$39.99	$39.99	$39 99
Hard Hat Mack	$27 99	$27 99	$27 99	$27 99
InfoCom				
Witness	$29 99	$29 99	$29 99	$29 99
Infidel	$29 99	$29 99	$29 99	$29 99
Deadline	$29 99	$29 99	$29 99	$29 99
Planetfall	$29.99	$29 99	$29 99	$29 99
Enchanter	$29 99	$29.99	$29.99	$29 99
Zork1,2,3 ea	$27.99	$27.99	$27.99	$27.99
Suspended	$29.99	$29.99	$29.99	$29.99
Sorcerer	$29 99	$29 99	$29.99	$29.99
AtariSoft				
Joust	$35 99	N A	$28 99	$28 99
Moon Patrol	$35 99	N A	$28 99	$28 99
Ms PacMan	$35 99	N A	$28 99	$28 99
PacMan	$35 99	N A	$28 99	$28 99
Donkey Kong	$35 99	N A	$28 99	$28 99
Pole Position	$35 99	N A	$28 99	$28 99
Spinnaker				
Aerobics	$28 99	$28.99	$28.99	$28.99
Trans	$24 99	$24.99	$24.99	$24.99
Adventure Creature	$24.99	$24.99	$24.99	$24.99
Aegean Voyage	$24 99	$24 99	$24.99	$24.99
Snooper Troops 1,2 ea	$28 99	$28 99	$28.99	$28.99
Traction Fever	$22 99	$22 99	$22 99	$22 99
Alphabet Zoo	$22 99	$22 99	$22 99	$22 99
In Search of	$24 99	$24 99	$24 99	$24 99
Facemaker	$22 99	$22 99	$22 99	$22 99
Kinder Comp	$17 99	$17 99	$17 99	$17 99
Dynatech				
Code Writer	$79 99	$79.99	$175.99	$155.99
VisiCorp				
VisiCalc	$159.99	$149.99	$159.99	$159.99
VisiCalc Advanced	N/A	N/A	N/A	$269.99
pfs:				
Write	N A	N A	$89 99	$89 99
Graph	N A	N A	$89 99	$79 99
Report	N A	N A	$79 99	$79 99
File	N A	N A	$89 99	$79 99
Solutions * as low as	N A	N A	$16 99	$16 99

*Call on Titles

SHARP

PC-1500A ... $165.99
PC-1250A ... $88.99

CE-125 Printer/Cassette ... $128 99
CE-150 Color Printer/Cassette ... $171.99
CE-155 8K RAM ... $93.99
CE-161 16K RAM ... $134.99

NEC

NEC

PC-8221A Thermal Printer ... $149 99
PC-8281A Data Recorder ... $99.99
PC-8201-06 8K RAM Chips ... $105.00
PC-8206A 32K RAM Cartridge ... $329.00

commodore

CBM 8032 ... $599
CBM 8096 ... $869 00
CBM 9000 ... $999 00
B128-80 ... $769 00
8032 to 9000 Upgrade ... $499 00
2031 LP Disk Drive ... $299.00
8050 Disk Drive ... $949.00
8250 Disk Drive ... $1199.00
4023 Printer ... $379.00
8023 Printer ... $569.00
6400 Printer ... $1399 00
Z-RAM ... $499.00
Silicon Office ... $699 00
The Manager ... $199 00
SoftROM ... $125 00
VisiCalc ... $159.00

PROFESSIONAL SOFTWARE

Word Pro 2 Plus ... $159.00
Word Pro 3 Plus ... $189.00
Word Pro 4 Plus/5 Plus each ... $279.00
InfoPro ... $179 00
Administrator ... $399 00
Power ... $79 99

SX-64 Portable ... $839
CBM 64 ... $199
C1541 Disk Drive ... $249 00
C1530 Datasette ... $69.99
C1520 Color Printer/Plotter ... $129 00
M-801 Dot Matrix Printer ... $219.00
C1526 Dot Matrix/Serial ... $299.00
C1702 Color Monitor ... $259.00
C1311 Joystick ... $4.99
C1312 Paddles ... $11.99
C1600 VIC Modem ... $59.99
C1650 Auto Mdoem ... $89.99
Logo 64 ... $49 99
Pilot 64 ... $39 99
Word Pro 64 Plus ... $59 99
Calc Result 64 ... $65 99
Calc Result Easy ... $39.99
Codewriter 64 ... $75.99
MCS 801 Color Printer ... $499.00
DPS 1101 Daisy Printer ... $459.00
Magic Voice Speech Module ... $54.99
Desk Organizer Lock ... $49.99
Vidtex Telecommunications ... $34 95

MSD

SD1 Disk Drive ... $349 00
SD2 Disk Drive ... $599 00

COMPUTER MAIL ORDER

VISA

west
800-648-3311
In NV call (702)588-5654, Dept. 125
Order Status Number: 588-5654
P.O.Box 6689
Stateline, NV 89449

canada
Ontario/Quebec 800-268-3974
Other Provinces 800-268-4559
In Toronto call
(416)828-0866, Dept. 125
Order Status Number: 828-0866
2505 Dunwin Drive, Unit 3B
Mississauga, Ontario, Canada L5L1T1

east
800-233-8950
In PA call (717)327-9575, Dept. 125
Order Status Number: 327-9576
Customer Service Number: 327-1450
477 E. 3rd St., Williamsport, PA 17701

MasterCard

CANADIAN ORDERS: All prices are subject to shipping, tax and currency fluctuations. Call for exact pricing in Canada. INTERNATIONAL ORDERS: All orders placed with U S offices for delivery outside the Continental United States must be pre-paid by certified check only. Include 3% (minimum $5) shipping and handling. EDUCATIONAL DISCOUNTS: Additional discounts are available to qualified Educational Institutions. APO & FPO: Add 3% (minimum $5) shipping and handling.

Question Mark Unraveled

Dear Drs. Ferra & Cortease:

After many hours of research, skipping meals, and closing down the library several times, I believe I have uncovered the secret to the North American Rosetta stone discovered by Drs. Ferra and Cortease.

The communications of the Toltecs and Aztecs using Pyramidems of the Moon, I found out, was somewhat slow due to the serial pottery interface required. This led to much experimentation and finally the development of the much more efficient Pyramodens of the Sun. These were parallel devices and required only minor modification to the Adobe software. With the new hardware, everyone had access to the Teotihuacan data bases.

A drawback still remained in data transmission. An Alpha Spatial Scan [ASS] was needed to provide proper displaying on the Ceramic Refined Tritons (CRTs). Plumed Serpentine Software, Inc., with main offices at the Temple of Quetzalcoatl, overcame the problem and provided the solution and test data to all who requested it.

The test data could be obtained on clay or the more reliable rockettes. It was in 126 position spiral form and was read from the outside inward, breaking on every seventh position creating 18 new positions. The graphic representations of the 18 positions was then reversed and message "WELCOME TO LYTE BYTES" was displayed.

Oliver H. Wardlow, Jr.
Topeka, KS

Editor's Note:
Our thanks to Dave Nicklas of Danvers, MA who was the first to send in the correct solution.

Unusual Software Sought

Dear Sirs:

I am interested in point-of-sale cash registers, bar code readers, and software for liquor store applications.

Cleo McCoy
Marietta, GA

Dear Sirs:

I am interested in Apple II Plus software and related hardware for use in quality control of nuclear fuel.

Hyun Tae Kim
Korea Advanced Energy Research Institute
Chung-Nam, Korea

Dear Sirs:

I am writing to ask if you know where I might be able to obtain hardware and software for an Apple II Plus to operate and control a fish nursery. I am very interested in any information about setting up and operation of computerized fish farms/nurseries.

Joseph G. Bloechl
APO New York

Editor's Note:
If any readers can suggest software to fit any of the above requests, please send name of package, publisher name and address to Letters Editor at Micro.

(continued)

More on the 68000

Gentlemen:

I read with interest Paul Lamar's article on the 68000 in your June issue. Like Paul, I am enthusiastic about the 68000. As I repeatedly read in that issue, 68000 will indeed be the 6502 of the 1980's. If I were more tactful, I would refrain from pointing out that I made exactly that assertion to Bob Tripp three years ago.

On page 45, Paul has an incomplete quote from the '*DTACK GROUNDED* newsletter. In its incompleteness, it appears to be wrong. Paul asserts that the 68000/16081 combination can perform a double precision multiply in 23 microseconds and that is 3 times faster than an 8086/8087 combination. Since most folks familiar with the 8087 know it performs a double-precision multiply in 27.4 microseconds, the figures do not appear to jibe.

A more complete explanation: the 12.5MHz 68000/6.25MHz 16081 can perform the double-precision operation A = B * C, where A, B and C are double-precision operands kept in memory, in 23 microseconds. A 5MH 8086/8087 takes 9 microseconds to load A, 9 microseconds to load B, 27.4 microseconds to perform the multiplication, and 20 microseconds (!) to store the result in memory: a total of 65.4 microseconds. That does not include the 'EA' (effective address) time, so a practical Intel system is in fact about 3 times slower than the 68000/16081 system. (The 23 microseconds for the 68000/16081 system represents an actual *measurement* of a loop repeated 1,000,000 times with the loop overhead subtracted.)

As you can see, the 8086/8087 system has a considerable overhead associated with transferring floating point operands to and from memory. The reason is that the data representation used by the 8087 internally is not the same which is stored in memory, and the conversion takes an appreciable amount of time. By way of contrast, the Nat Semi 16081 math chip uses an internal data representation which is identical to that which is stored in memory. Also, the actual multiplication takes place much faster - 10 microseconds vs. 27.4.

There are other differences between the 8087 and the 16081, such as the fact that the 8087 does most transcendental calculations as a single command and the 16081 does not, so the 16081 is going to be a lot faster than the 8087 when performing linear algebra or matrix math (where most operands are kept in memory) while the 8087 will be faster than the 16081 when calculating square roots or arc tangents. Nothing is simple these days, is it?

A very limited number of samples of 8 MHz 8087s exist and so does a very limited number of samples of 10MHz 16081s. As far as us peasants are concerned, neither part really exists right now.

Finally, the application note which Nat Semi is preparing shows a 68000 slowed down to the same clock speed as the 16081, which is the way the Nat Semi 16032 microprocessor has to work with the 16081. Nat Semi does not want anybody to know about the way *we* use the 16081, running at half the clock speed of the 68000 because such a system can outrun the Nat Semi combo when nonfloating point operations are being performed.

For the record, Digital Acoustics shipped the first commercial 68000/16081 system in Dec. '83. To the best of our knowledge, we are the only ones actually shipping product today. The fact that lots of folks are actively working on such systems may have something to do with the fact that we published a schematic and a four-page technical explanation in our newsletter 25, (10/22/83).

Hal W. Hardenbergh
President, Digital Acoustics, Inc.

spotlight

Micro-Match Interface Series

Distributor

Command Computer Corporation
P.O. Box 5096
Philadelphia, PA 19111

Introduction

In this day and age of such a varied host of computers, peripherals and interfaces, one is left in a state of confusion as how to get them all together. Command Computer Corporation's 'Micro-Match' is a great step forward in helping to resolve this communication problem. It has been designed to 'take the guesswork and inefficiency out of interfacing or interconnecting.' What Micro-Match supports is micros to printers, CRT's, modems and plotters; micros to minis; and micros to mainframes. The product comes in two parts; a two volume product called Blueprints and a second product called Access Forum.

Blueprints

Blueprints is a set of ready-to-use interfaces. There are indices by manufacturer and device type. These indices direct you to sections that list wiring specs, switches and jumpers, and any other pertinent information. Cross-referencing makes it easy and fast to find the specifications and information you need to interface two products. In the Wiring Specs section whether an interface is parallel or serial is confirmed, and a diagram is included showing the actual connections that need to be made on both devices. Switches and jumpers is the other major section necessary in interfacing. This part clearly shows you how to configure your devices using diagrams and notes. The section entitled General Notes is gold mine of information that has been gathered into a handy and readable form. Covering grounding principles, cable length limits, cable wire types, checklists, plug types, and more, it is an invaluable aid in telecommunications. Also provided are Device Notes which show diagrams of the actual units, where the various ports, buttons and controls are located, showing all necessary views for a complete picture. Updating supplements will be provided to each Micro-Match user every quarter. These will keep the user up-to-date with new products and their interfaces.

Some of the specific computers and peripherals covered are: Altos, Apple, AST Research, California Computer Systems, IBM, Kaypro, Quadram, Tecmar, Televideo, Hayes, Novation, Anadex, C-Itoh, Comrex, Diablo, NEC, Epson, Okidata, and more. Additionally, users will be able to request 'custom interfaces' for devices not covered in Blueprints.

Access Forum

The second product, Access Forum, is 'an on-going series of research reports that focus on important products and topics related to the interfacing of microcomputers and peripherals with mini and mainframe computers.' This section is not for the uninitiated but rather is designed with the data processing manager and professional in mind. It is an aid to these people who are searching for specific solutions to their interfacing problems in this area. The solutions that are focussed on are in the following areas: downloading/uploading data files between micros, minis, and mainframes; converting mini and mainframe software to micros; interfacing micros with mainframe-compatible tape drives, card readers, and special input/output devices; connecting micros with TELEX and commercial timesharing networks; micro upgrade paths; converting/porting programming languages; networking micros; converting protocols; interfacing graphics devices, graphics systems compatibility, etc.; integrating micros into 3270, SDLC, and SNA networks. For the users convenience two copies of the Forum are provided which can be easily removed and kept handy.

Pricing

Both Blueprints and Access Forum can be purchased for $690 per year. Beyond the initial products the price includes 'quarterly Blueprints supplements with interfaces for devices to be announced in the industry during the coming year, and three new Access Forum volumes on new micro-to-mini and micro-to-mainframe access products and solutions.'

Conclusion

The need for such a product is clear given the ever increasing number of micros, minis, peripherals, etc. The problems associated with telecommunications are often aggravating, time-consuming and difficult. Micro-Match does indeed fill a nitch that has been neglected. And it is what it says it is 'a complete, step-by-step system that can be used by anyone having minimal computer experience to successfully implement a complete working interface between a wide variety of microcomputer devices and peripherals.' The Access Forum also is an invaluable source of state-of-the-art information as regards micro, mini, and mainframe interfacing. Combined both Blueprints and Access Forum are an investment that can save time, money and unnecessary problems.

Product Name: **When I'm 64**
Equipment Req'd: Commodore 64 and disk drive
Price:
Manufacturer: The ALIEN Group
27 West 23rd St.
New York, NY 10010

Description: A sophisticated music synthesizer music system including vocal parts and an animated face. The disk is supplied with 30 short demo songs and a short instruction manual for generating your own music. With the addition of the ALIEN Group Voice Box, an graphics screen face also sing the lyrics of the song. In addition to all of the standard controls (envelope, etc.) the program also has controls for vibrato, glissando, and accents. Lyrics are entered in the phonetic alphabet.

Pluses: A screen editor is supplied to easily modify any of the synthesizer (sound chip) parameters, with immediate playback and correction of input songs. The notes and words are easily edited and the face animated to imitate the lyrics.

Minuses: Any program of this magnitude is going to be difficult to learn to use all its facets without much study.

Skill level required: No previous background assumed.

Reviewer: Phil Daley

Product Name: **BASIC Building Blocks**
Equipment Req'd: Apple II,II+,IIe - DOS 3.3 (ProDOS available). Also available for Atari and C-64.
Manufacturer: Micro Education Corp. of America
285 Riverside Avenue
Westport, CT 06880

Description: Disk 1: An interactive tutorial in the BASIC language using over 60 sample programs describing all facets of beginning programming -I/O, branching, arrays, strings, graphics and disk access. Disk 2: Basic Design Tool, a M/L runtime debugging tool to help in understanding how programs work or why they don't work. This program is worth the price of the package alone.

Pluses: While the sample programs are good for the beginning programmer, the BDT really makes the package interesting. You can set breakpoints by line number or variable name; run, trace or step the program; and switch between the program's screen display and the debugger display which includes preset variables and their values, the just executed statement and the next statement to be executed, and stack information about FOR..NEXTs and GOSUBs. The documentation, while short is almost unecessary due to program friendliness.

Minuses: BDT does take up 8K space (above HIMEM) and pushes the BASIC program above Text Page 2 (it uses Page 2 for its own display), so not all BASIC programs will fit in the leftover space. PEEKing the keyboard ($C000) also won't work, although you can RUN past those points.

Skill level required: No previous knowledge for Disk 1. Some BASIC programming experience for Disk 2.

Reviewer: Phil Daley

Product Name: **Write Now!**
Equipment Req'd: Commodore 64 with disk or tape and printer
Manufacturer: Cardco, Inc.
313 Mathewson
Wichita, KS 67214

Description: A professional word processor with 80 column display for sample output, unlimited length documents, search and replace, multiple line headers and footers, justification, block copy and delete and compatibility with almost any printer. You can select screen colors, view help screens, get a disk directory, initialize a disk, rename or scratch files and more, all from within Write Now!.

Pluses: Unless the file is very large (more than 16K), the program is very fast, since it is entirely in machine language. Usable memory is about 30K, but files can be chained together. The documentation is excellent with a very good index to find answers to any questions. There is also a cut-out card to indicate what the control/ Commodore functions are.

Minuses: There is no word wrap! Words are broken at the end of the lines. You may be able to tell where you are in 80 column display, and you may not. It depends on the CRT. The search and replace does not function if you try to replace a character with nothing. There appears to be a bug - if you have a full file and replace something with a longer something, it will replace several before stopping, creating a file longer than allowed and full of gibberish at the end. Reloading a that saved file gives a Memory overflow error, which throws away all the gibberish at the end. The manual says that a joystick will move the cursor; mine didn't.

Skill level required: No previous experience required.

Reviewer: Phil Daley

Product Name: **Songwriter**
Equipment
Req'd: Commodore 64 and disk drive
Price:
Manufacturer: Scarborough Systems, Inc.
25 North Broadway
Tarrytown, NY 10591
Author: Samuel Wantman

Description: An educational program that makes it fun to learn to write songs. Songs are recorded in a piano-roll style with simple editing commands to alter, add or delete notes. Songs can be saved on disk with 20 songs prerecorded. Tempo is variable; songs can be stopped anywhere and single stepped forwards or backwards. Sound quality is somewhat variable using the function keys.

Pluses: Entering songs is extremely simple whether entering from sheet music or original material. The duration is set (1/4, 1/8th note, etc.) and the pitch selected from a graphic keyboard with cursor keys or joystick. Playback can be immediate. The thorough documentation is easy to read and very complete. This program can easily used by young children.

Minuses: The program only allows single voice parts. Unless you are interested in melody only, or have an Apple, this drawback defeats most of the usefulness of the Commodore sound chip. It also only allows limited sound adjustment.

Skill level required: No prior knowledge needed.

Reviewer: Phil Daley

Product Name: **Clone Master**
Equip. Req'd: TRS-80 Color Computer,
16K Disk Basic
Price: $34.95
Manufacturer: Prickly-Pear Software
9234 E. 30th Street
Tucson, AZ 85710

Description: Clone Master is a disk utility which is designed to create backup copies of all or selected portions of user-owned diskettes. It is not designed or intended for use in the illegal copying of copyrighted software. It will run on 16K, 32K or 64K Color Computers with at least one disk drive. The program will check the RPM of the disk drive before beginning any backup work. If the drive is not within acceptable parameters, an error message will be issued to the effcct that the drive speed needs adjusting. Double sided drivcs are also supported and backup can be done from one side to the other. Any range of disk tracks from 0 to 99, inclusivc, may be backed up. Backup of partial diskettes is, therefore, possible.
⅓
Pluses: Clone Master will copy everything that is on a disk, including errors. It can be used to copy non-Color Computer diskettes as well, which makes it handy as a utility if you have other types of computers in addition to the Color Computer. With 64K machines, the backup process uses the entire memory available to perform the backup, necessitating only three disk swaps for single disk drives as opposed to seven swaps for the Disk Basic Backup command. The RPM check feature is a good diagnostic test. Track step rates can be changed and the program itself can be custom tailored to a user's configuration and then permanently saved to disk for future use. Partial copying is a good feature and allows the user to copy only selected portions of a diskette.

Minuses: The user must know exact contents of a disk if partial backup is to be used. There is no provision in Clone Master to view the contents of any disk or the directory.

Documentation: The documentation consists of three typewritten pages; it is adequate to use the program. All features are discussed and described in the documentation and additional technical information is presented for those who may wish to modify the program for tailored execution.

Skill level: The program can be used by novice users with relative ease. It is completely menu driven and error messages are explained in depth in the documentation.

Reviewer: Norman Garrett

Product Name: **Advanced System Editor**
Equipment
Req'd: Apple II,II+,IIe - Apple Pascal
Manufacturer: Volition Systems
P.O. Box 1236
Del Mar, CA 92014
Author: Richard Gleaves

Description: ASE is an extended Pascal SYSTEM.EDITOR running under the Apple UCSD Pascal system. While fully compatible with the standard editor, including using all of the standard features and commands, it includes many enhancements and additional features making it much more versatile and easy to use. Files are not limited to memory size and may be as large as an entire disk. A disk directory, including the first line of the file, is available from within the editor. Multi-file editing is possible, making it easy to move sections from one file to another. ASE also has user definable function keys.

Pluses: The editor is much faster and has many more commands making text and program editing easier and more efficient. A backup of the original file is always saved in case of mistakes. It has a column command to move whole columns left or right. Installation is easy and the documentation is superb. If you write many Pascal programs, you need this editor.

Minuses: None noted.

Skill level required: Familiarity with UCSD Pascal.

Reviewer: Phil Daley

MICRO

Product Name: **Total Health**
Equipment Req'd: Commodore 64 and disk drive
Manufacturer: Computer Software Associates
65 Teed Drive
Randolph, MA 02368
Author: Mark Baier

Description: A program for fitness and health enthusiasts which monitors and encourages good nutrition. The program has two parts: a file manager of daily food intake which keeps track of the calories, protein, fat and carbohydrates consumed on a daily basis; and a graphing program to plot your current status toward your final goal. The package includes a small manual that is not really needed due to the program's elegant simplicity.

Pluses: The program is easy to use and can be your own personal weight watcher. Like any dietary aid, it will encourage you to do well, but the hard part is to stick to your schedule. The list of foods could be larger, but there is a provision for adding your own values for unlisted foods.

Minuses: There aren't many input checks, so the program crashes relatively easily. I had several problems with the disk routines.

Skill level required: No prior knowledge necessary.

Reviewer: Phil Daley

Product Name: **The Commodore 64 Music Master**
Equipment Req'd: Commodore 64 and cassette
Manufacturer: Softtext, Inc.
Cambridge, MA
Author: James Vogel

Description: A tape of over 50 BASIC music programs for the C-64 with accompanying book describing how the programs work, variable listings and logic flow, utilities for exploring the ADSR, filters and frequency conversions, and suggestions for making your own music programs. The programs themselves are generally interesting and range from simple to complex. The documentation procedes from a very beginning sound program, stepwise to more advanced topics.

Pluses: The programs are well documented and can easily be modified to help in understanding how they work. They include a wide range of topics and show how to use all of the features available in the SID chip. It is an excellent introduction to the sound capabilities of the C-64 in BASIC.

Minuses: None noted.

Skill level required: A small amount of programming experience (1 or 2 weeks) to be able to modify the programs.

Reviewer: Phil Daley

Product Name: **Magic Memory**
Equipment
Req'd: Apple II,II+,IIe - DOS 3.3
Manufacturer: ARTSCI
5547 Satsuma Ave.
North Hollywood, CA 91601
Author: Executive Software, Inc.

Description: An address-book oriented data-base system utilizing either 40, 70 or 80 column screen display and allows up to 9 lines of information per entry. Entries are saved under a TAB name - there are 48, 24 alphabetic and 24 user defined. Each set of TAB entries is saved in its own text file. TAB files may be sorted on any field, left or right hand sort, and the sorted file may replace the original.

Pluses: The documentation is very good and the program is very easy to use, especially for a novice. For names, addresses and similar short entry type of information, the program is very good. It has flexible printouts and entries can be easily replicated into other tab files for cross-reference ability without retyping.

Minuses: The program has a limited scope of useage. The field format is preset and unchangeable. There is no searching ability to find a particular record, other than flipping through records one at a time.

Skill level required: No previous knowledge necessary.

Reviewer: Phil Daley

Product Name: **64 Doctor**
Equipment
Req'd: Commodore 64 and disk drive
Manufacturer: Computer Software Associates
44 Oak St.
Newton Upper Falls, MA 02164
Author: Eric Berkowitz & David Pollack

Description: A diagnostic aid to troubleshoot your C-64 including RAM, keyboard and audio, and associated peripherals - disk drive, printer, cassette, joystick, and video unit. They can be tested individually or in an auto-test mode. Any problems encountered are printed on the screen.

Pluses: The disk and short instruction manual are a lesson in user friendliness. The program uses an interesting graphic presentation to simplify use. Problems are clearly spelled out with a suggestion to take the offending equipment to a repairman.

Minuses: If the C-64 is not working or the disk drive won't load a program, you'll never get this program in to determine what's wrong. Saving a copy on a cassette would probably be a good idea.

Skill level required: No prior knowledge necessary.

Reviewer: Phil Daley

Product Name: **The Computer Mechanic**
Equipment
Req'd: Commodore 64 and Disk Drive
Manufacturer: Softsync, Inc.
14 East 34th St.
New York, NY 10016

Description: This program is a computerized automotive instruction and diagnostic tool. It teaches an auto novice about possible causes of various problems and the recommended time intervals between certain checks and maintainence. Covered topics include oil change, brake job, transmission check, tune up and tire rotation. The program will also save a file on up to 100 different vehicles, to help keep track of the last performed maintainence dates.

Pluses: The program is extremely easy to use and requires almost no instruction manual. There are interesting graphics of the various parts of a car.

Minuses: If you know almost anything about cars, this program is too simple to be of any value. The save the date section of maintainence might be useful if the program looked to see when maintainence is due, but it doesn't. You have to look at the individual records and then write them down while you look at the maintainence interval section to see if any is due. No printer output.

Skill level required: None

Reviewer: Phil Daley

A Relocatable Step/Trace

by Chester H. Page

Step & Trace program can be moved easily to any part of your program.

Listing 1

```
0022          W EQU $22
002F          LGTH EQU $2F
0033          PRMP EQU $33
0034          YSAV EQU $34
003A          PCL EQU $3A
003B          PCH EQU $3B
003C          XQT EQU $3C
0048          STAT EQU $48
0100          STCK EQU $100
0670          UW EQU $670
0674          TW EQU $674
C000          KBRD EQU $C000
F882          INSD EQU $F882
F8D0          DISA EQU $F8D0
F954          ADJ2 EQU $F954
F956          ADJ3 EQU $F956
FAD7          REGD EQU $FAD7
FADA          RGDS EQU $FADA
FC22          VTAB EQU $FC22
FD67          GETL EQU $FD67
FDED          COUT EQU $FDED
FE00          BL1 EQU $FE00
FE75          A1PC EQU $FE75
FF3A          BELL EQU $FF3A
FF3F          RSTR EQU $FF3F
FF4A          SAVE EQU $FF4A
FFA7          GETN EQU $FFA7
FF58          RTRN EQU $FF58
FFBE          TSUB EQU $FFBE
FFC5          TSB1 EQU $FFC5
FFC7          ZMOD EQU $FFC7
FFCC          CHRT EQU $FFCC
              *
              *INITIALIZE WINDOW AREAS
              *
                    ORG $7000
                    OBJ $800
7000  A9 00         LDA #0
```

MICRO on the Apple, Vol. 2, contains a Step and Trace program for the Apple II+; it is an adaptation of the step and trace routines in the old monitor ROM, so has the same problems. The article warns that DOS causes trouble when it changes an output hook, as in the COUT routine, but there is no warning about the other problems I encountered on my first attempt to use the program.

The user program I wanted to trace contained a printing routine of the type:

```
        LDX #TEXT
        LDY /TEXT
        JSR PRINT
PRINT   STX PTR
        STY PTR+1
        LDY #0
LOOP    LDA (PTR),Y
        BEQ DONE
        JSR COUT
        INY
        JMP LOOP
DONE    RTS
```

where TEXT is the address of the data: 8D D4 C5 D8 D4 00. This asks to print a carriage return (for a fresh line), the word "TEXT" and to leave on reading the zero byte.

The DOS problem is easily solved by BLOADing STEP/TRACE, then disconnecting DOS by entering PR#0, IN#0.

My first unexpected problem was

```
7002  85 22           STA W
7004  A9 14           LDA #$14
7006  85 23           STA W+1
7008  A9 13           LDA #$13
700A  85 25           STA W+3
700C  A9 16           LDA #$16
700E  8D 70 06        STA UW
7011  A9 18           LDA #$18
7013  8D 71 06        STA UW+1
7016  A9 00           LDA #0
7018  8D 72 06        STA UW+2
701B  A9 17           LDA #$17
701D  8D 73 06        STA UW+3
                *____________________________
7020  D8        STRT  CLD
7021  20 3A FF        JSR BELL
7024  A9 2A     CONT  LDA #$2A
7026  85 33           STA PRMP
7028  20 67 FD        JSR GETL
702B  20 C7 FF        JSR ZMOD
702E  20 A7 FF  NXTI  JSR GETN
7031  84 34           STY YSAV
7033  C9 EC     TRYS  CMP #$EC
7035  F0 0B           BEQ ENT2
7037  C9 ED     TRYT  CMP #$ED
7039  D0 0D           BNE TRCR
703B  AD 00 C0        LDA KBRD
703E  30 22           BMI AGIN
7040  C6 34           DEC YSAV
7042  20 C7 FF  ENT2  JSR ZMOD
7045  B8              CLV
7046  50 21           BVC STPZ
7048  C9 C6     TRCR  CMP #$C6
704A  D0 09           BNE MCMD
704C  20 C5 FF        JSR TSB1
704F  20 00 FE        JSR BL1
7052  B8              CLV
7053  50 CF           BVC CONT
7055  A0 17     MCMD  LDY #$17
7057  88        CHRS  DEY
7058  30 C6           BMI STRT
705A  D9 CC FF        CMP CHRT,Y
705D  D0 F8           BNE CHRS
705F  20 BE FF        JSR TSUB
7062  A4 34     AGIN  LDY YSAV
7064  B8              CLV
7065  50 C7           BVC NXTI
7067  50 B7     RS3   BVC STRT
7069  20 75 FE  STPZ  JSR A1PC
706C  20 D0 F8  STEP  JSR DISA ; DISPLAY INSTRUCTION
706F  A2 02           LDX #2
7071  A9 EA     XQIN  LDA #$EA
7073  95 3C           STA XQT,X
7075  CA              DEX
7076  D0 F9           BNE XQIN
                *____________________________
                *INITIALIZE EXECUTE AREA WITH NOP'S AND RETURN JUMPS
                *____________________________
7078  20 58 FF        JSR RTRN
707B  B8              CLV
707C  50 0E           BVC BR
707E  18        BRAN  CLC
707F  A0 01           LDY #1
7081  B1 3A           LDA (PCL),Y
7083  20 56 F9        JSR ADJ3
7086  85 3A           STA PCL
7088  98              TYA
7089  38              SEC
708A  B0 6C           BCS R2 ; JUMP TO PCN2
708C  BA        BR    TSX    ; LOCATE BRAN AND INSERT JUMP COMMAND
```

caused by the carriage return ($8D). Somehow this gets repetitively tangled with the returns in the disassembly routine, with the result that the screen is scrolled with a complete set of blank lines and the trace program left in an infinite loop of scrolling commands! Removing the 8D uncovered a second problem. In COUT, Y is saved (in YSAV1, $35) and restored after the actual printing is performed. But each of the intermediate steps is processed by Step-Trace, using COUT to display these steps. The net result is that when the COUT routine in the user program is finished, it always restores Y to 0 and the second letter of text (Y = 1) is printed repetitively.

The third problem is that the letters that do get printed overwrite a character in the command display, so that they are not evident. I decided to modify Step-Trace to eliminate these problems.

The first two were cured by examining each command to see if it is "JSR COUT". When this is encountered, the COUT routine is called directly by my trace program so that it will not be traced stepwise. This also eliminated the problem with DOS; DOS no longer needs to be disconnected. The third problem is cured by establishing a split screen; whenever the substitute COUT is to be called, the window parameters being used by the trace program (window top, bottom, horizontal and vertical cursor positions) are saved and replaced by a user set. After a text-character printout, this user set is saved and the trace-program set replaced. With this modification, the TEXT is printed out properly in the two bottom screen lines and all program display is kept above it (with a blank gap between).

All these improvements lengthen the program so that it will no longer fit in the $300 page. Since it should be useful for tracing a program located anywhere, it should be written in relocatable form so that it can be used by BRUN STEP/TRACE, AX where X is any convenient location that avoids the program to be examined. The internal jumps were converted to relative jumps using CLV followed by BVC and relay points inserted when the jump distances were too large. The

only real problem was figuring out how to convert the routine for installing the jump commands that follow the copied user-program command at $3C, since the commands to be transferred are themselves internal program jumps (to NBRN and BRAN, for returns from "no-branch" and "branch" operations).

I solved this problem by eliminating these jump instructions from the trace program and installing the NBRN and BRAN addresses directly into the jump commands at $3F/44. This is done by using a location finding routine at BRAN and again at NBRN.

I first tried doing this with BRAN and NBRN at the beginning of the program so that the self-locating routine wouldn't have to be repeated with every user command, but could be used once as part of the initialization. Unfortunately, some of the monitor subroutines such as GETN in NXTI use some of the memory locations right after $3C and the initialization gets overwritten. My second try was to move the XQT area from $3C to $E0, since this would be interfered with only by HiRes graphics. This worked fine, but then I realized that the $E0 area is sometimes used for program pointers in non-graphics programs, so that there could be interference. For this reason, I finally put NBRN and BRAN in the XQIN area, where the $3C region is initialized before each program command.

The final decision to be made concerned the storage of window parameters. Again, zero page is out because of possible interference with user programs. There are nice unused areas available up in DOS, but again there can be interference; I have used many of these locations for data storage in long complex programs that exhaust zero page.

For a trace program at a fixed location the window parameters could be stored within the program, but I was soon convinced that relocatability would be too complicated. My final choice was to dedicate one screen line to this storage. Since I wanted a blank area to separate the trace window from the user window, I put the window parameter storage at the end of one of

```
708D  CA              DEX
708E  18              CLC
708F  BD 00 01        LDA STCK,X
7092  69 04           ADC #4
7094  85 43           STA XQT+7
7096  E8              INX
7097  BD 00 01        LDA STCK,X
709A  69 00           ADC #0
709C  85 44           STA XQT+8
709E  A9 4C           LDA #$4C
70A0  85 3F           STA XQT+3
70A2  85 42           STA XQT+6
70A4  20 58 FF        JSR RTRN
70A7  B8              CLV
70A8  50 0A           BVC NB
70AA  20 4A FF NBRN   JSR SAVE ; SAVE USER REGISTERS
70AD  38              SEC
70AE  B0 4A           BCS R3   ; JUMP TO PCN3
                 *______________________________
70B0  50 B5    RS2    BVC RS3   RELAY
70B2  50 AE    RA3    BVC AGIN  RELAY
                 *______________________________
70B4  BA       NB     TSX       ; LOCATE NBRN AND INSERT JUMP COMMAND
70B5  CA              DEX
70B6  18              CLC
70B7  BD 00 01        LDA STCK,X
70BA  69 04           ADC #4
70BC  85 40           STA XQT+4
70BE  E8              INX
70BF  BD 00 01        LDA STCK,X
70C2  69 00           ADC #0
70C4  85 41           STA XQT+5
                 *______________________________
                 * COPY USER COMMAND TO XEQ AREA
                 *______________________________
70C6  A2 00           LDX #0
70C8  A1 3A           LDA (PCL,X)
70CA  F0 3A           BEQ XB1  ; JUMP TO XBRK
70CC  A4 2F           LDY LGTH
70CE  C9 20           CMP #$20
70D0  F0 38           BEQ CHECK
70D2  C9 60           CMP #$60
70D4  F0 32           BEQ XR1    ; JUMP TO XRTS
70D6  C9 4C           CMP #$4C
70D8  F0 28           BEQ X1
70DA  C9 6C           CMP #$6C
70DC  F0 26           BEQ XT
70DE  C9 40           CMP #$40
70E0  F0 76           BEQ XRTI
70E2  29 1F           AND #$1F
70E4  49 14           EOR #$14
70E6  C9 04           CMP #4
70E8  F0 02           BEQ XQ2
70EA  B1 3A    XQ1    LDA (PCL),Y
70EC  99 3C 00 XQ2    STA XQT,Y
70EF  88              DEY
70F0  10 F8           BPL XQ1
70F2  20 3F FF        JSR RSTR     ; RESTORE USER REGISTERS
70F5  4C 3C 00        JMP XQT
                 *______________________________
70F8  B0 66    R2     BCS PCN2     ; RELAY
70FA  B0 66    R3     BCS PCN3     ; RELAY
70FC  50 AC    RN     BVC NBRN     ; RELAY
70FE  50 B0    RS1    BVC RS2      ; RELAY TO STRT
7100  50 B0    RA2    BVC RA3      ; RELAY TO AGIN
7102  F0 73    X1     BEQ XJMP     ; RELAY
7104  F0 72    XT     BEQ XJAT     ; RELAY
7106  F0 45    XB1    BEQ XBRK     ; RELAY
7108  F0 52    XR1    BEQ XRTS     ; RELAY
                 *______________________________
```

```
71ØA  AØ Ø1     CHECK  LDY #1        ;IS COMMAND COUT?
71ØC  B1 3A            LDA (PCL),Y
71ØE  C9 ED            CMP #$ED
711Ø  DØ 5A            BNE XJSR
7112  C8               INY
7113  B1 3A            LDA (PCL),Y
7115  C9 FD            CMP #$FD
7117  DØ 53            BNE XJSR
7119  A2 Ø3            LDX #3        ; USER COMMAND IS COUT
711B  B5 22     SVT    LDA W,X       ; SAVE TRACE WINDOW
711D  9D 74 Ø6         STA TW,X
712Ø  CA               DEX
7121  1Ø F8            BPL SVT
7123  A2 Ø3            LDX #3
7125  BD 7Ø Ø6  LDU    LDA UW,X      ; LOAD USER WINDOW
7128  95 22            STA W,X
712A  CA               DEX
712B  1Ø F8            BPL LDU
712D  2Ø 22 FC         JSR VTAB      ; POSITION CURSOR
713Ø  2Ø 3F FF         JSR RSTR      ; RESTORE USER REGISTERS
7133  2Ø ED FD         JSR COUT      ; PRINT IN USER AREA
7136  A2 Ø3            LDX #3
7138  B5 22     SVU    LDA W,X       ; SAVE USER WINDOW
713A  9D 7Ø Ø6         STA UW,X
713D  CA               DEX
713E  1Ø F8            BPL SVU
714Ø  A2 Ø3            LDX #3
7142  BD 74 Ø6  LDT    LDA TW,X      ; LOAD TRACE WINDOW
7145  95 22            STA W,X
7147  CA               DEX
7148  1Ø F8            BPL LDT
714A  B8               CLV
714B  5Ø AF            BVC RN        ; JUMP TO NBRN
714D  2Ø 82 F8  XBRK   JSR INSD
715Ø  2Ø DA FA         JSR RGDS
7153  B8               CLV
7154  5Ø A8            BVC RS1       ; JUMP TO STRT
7156  5Ø A8     RA1    BVC RA2       ; RELAY TO AGIN
7158  18        XRTI   CLC
7159  68               PLA
715A  85 48            STA STAT
715C  68        XRTS   PLA
715D  85 3A            STA PCL
715F  68               PLA
716Ø  85 3B     PCN2   STA PCH
7162  A5 2F     PCN3   LDA LGTH
7164  2Ø 56 F9         JSR ADJ3
7167  84 3B            STY PCH
7169  18               CLC
716A  9Ø 14            BCC NEWP
716C  18        XJSR   CLC
716D  2Ø 54 F9         JSR ADJ2
717Ø  AA               TAX
7171  98               TYA
7172  48               PHA
7173  8A               TXA
7174  48               PHA
7175  AØ Ø2            LDY #2
7177  18        XJMP   CLC
7178  B1 3A     XJAT   LDA (PCL),Y
717A  AA               TAX
717B  88               DEY
717C  B1 3A            LDA (PCL),Y
717E  86 3B            STX PCH
718Ø  85 3A     NEWP   STA PCL
7182  BØ F3            BCS XJMP
7184  2Ø D7 FA         JSR REGD      ; DISPLAY USER REGISTERS
7187  B8               CLV
7188  5Ø CC            BVC RA1       ; JUMP TO AGIN
```

these blank lines. (It shows as 8 inverse characters at the right.)

The final program still suffers from a very slight defect: it will not tolerate PR#1 or a machine loading of CSWL to activate the printer.

To test this Step/Trace program, (1)RUN TEST, enter FF58S CR , then S CR ,.... or (2) HOME, enter the TEXT program below, BRUN STEP/TRACE,A12345, enter 3ØØS CR , S CR ,.... or 3ØØT CR .

```
1Ø REM TEST

2Ø PRINT CHR$(4)
   ''BRUN STEP/TRACE'':
PRINT ''HELLO''

3Ø END

TEXT

CALL - 151

3ØØ:A2 18 AØ Ø3 2Ø Ø7 Ø3 86 Ø6
84 Ø7 AØ ØØ B1 Ø6 FØ Ø6 2Ø ED FD
C8 1Ø F6 ØØ 8D D4 C5 D8 D4 ØØ
```

Constructing Truly 3-D Mazes

by Dr. Alan Stankiewicz

Get a rat's eye view of the maze corridors as you walk through them.

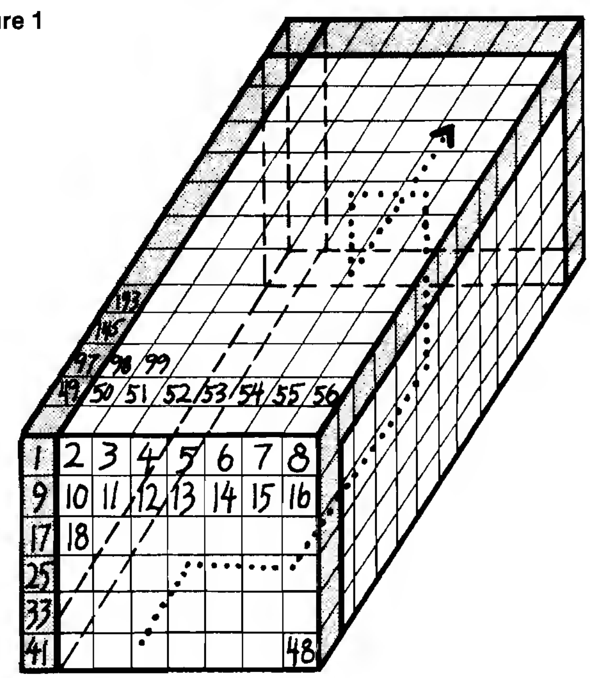
Just a few months ago, I remember reading an article in this journal concerning the construction of random mazes. The methods described were all quite interesting, however, they all had one thing in common - they were two-dimensional. Today, I will demonstrate a method of generating truly 3-dimensional random mazes on your computer with a minimum of effort and memory. To do this, I will be referring to a program called "Space Maze", written on the unexpanded VIC-20, reprinted here with permission from Victory Software Corporation. To emphasize the fact that not much memory is needed, this particular program will not only generate and store the maze but will actually give you a 3-dimensional rat's-eye view of the corridors as you walk through -all in 3 1/2K of RAM!

To start, you must envision a solid cube which is made up of a large number of smaller cubes, each of which are numbered according to the scheme shown in figure 1. Each of these smaller cubes will correspond to a byte in your computer's memory such that a maze 6 high, 8 wide and 13 deep will take up a block of memory 624 bytes long. This block may be stored anywhere in RAM but in this program, it begins at location 6751 and ends at 7375. The initial construction of this cube is done in line 9 by simply poking the number "2" into each of these bytes, indicating that they now represent a solid section of the maze.

The next major step in the creation of this maze is to wormhole our way through this cube from the front wall to the back wall in a random fashion, thus creating the main pathway. However, before this can be accomplished, we must further organize the large cube so that as we are drilling our way through, we will have some way of recognizing when we reach an outside wall and when we reach the exit (back wall). To accomplish this, we poke 4's into the bytes which represent the back wall and 3's into the bytes representing the left side, front and top walls (lines 9 and 30). Note that this procedure is not necessary for the right side or bottom wall (the reason will become apparent later).

Now that the limits of the cube are clearly marked, the drilling process may begin. The starting location for the main path in this program is selected to be 6808 which is the byte (or small block) located one layer down from the top, one layer in from the front and one layer over from the left (this would be block number 58 in figure 1). The number "1" will be poked in this location to show that it is now a "drilled out" byte and the variable "L" is set equal to 6808 (our present location). From here, we choose a random direction to start moving, but before drilling in that direction, we must peek the location directly in front of our drill to make sure it is not an outside wall (designated by a 3). If it is, another direction will have to be chosen. If it is an inside solid portion

Figure 1



(designated by a 2) then we will procede to drill out two bytes in that direction by poking 1's there and reassigning "L" equal to the location of the most freshly drilled out byte. The reason for moving two bytes at a time is so there will always be a wall between every corridor. After each drilling, the decision is made whether or not to change directions. For a more complex maze, of course, you would want to change directions more frequently. When you finally peek a "4", you know that you hit the back wall and your main path is complete, front to back. There is one difficulty which exists with this method, however, and that is the possibility of drilling yourself into an area which is totally surrounded by 3's and/or 1's with no possibility of finding a "2" (an undrilled inside byte) no matter which direction you look. This situation is easily resolved, however, by allowing yourself to backtrack over the main path every so often and as you do this, to fill up these "dead ends" with 3's (line 58) so you will never go there again. Backtracking can be kept to a minimum by only allowing it to occur, for example, after every 20th direction change.

In addition to controlling the frequency of direction changes, the difficulty level of the maze can also be regulated by setting a minimum length for the main path, that is, if in the drilling process you peek a "4", check the length of the main path (line 55). If it is not yet long enough, then change directions and continue to drill. The length of the main path is continuously monitored in "Space Maze" by adding 2 to the variable "U" every time you drill (line 54). Also don't forget to subtract from U when backtracking.

We now have a cube with a wormhole bored through it from front to back. The next major step, of course, is to create side paths to confuse the mouse. This task is accomplished in exactly the same manner as the main path, only we now start at randomly selected bytes on the main path and continue for variable distances outward. If memory permits, you may even want to differentiate the side paths by poking a "5" into these bytes, thereby creating a method of distinguishing the main path from the side paths during the display of the maze.

This brings us to the final problem of how to display our truly 3-D maze on a 2-D television screen. One technique is to display successive layers of the maze, as if you were able to slice it up with a knife and show one slice at a time (figure 2). Another more dramatic representation is to let the user "walk" through the maze, giving him a 3-dimensional view of the passageways as shown in figures 3, 4 & 5. At first glance, the production of this type of display would seem quite complicated, but in principle, all that is required is to draw a big "X" on the TV screen to represent a long hallway and then fill in the doorways as you scan that particular area of the maze. The detailed mechanics of this process are beyond the scope of this article, however, a BASIC program written fairly efficiently can usually draw up such a picture in 2-3 seconds.

I hope this short discussion has succeeded in arousing some interest in maze-building and I would be very interested to hear from anyone who has been able to generate and/or display a 4-dimensional maze!

Figure 2

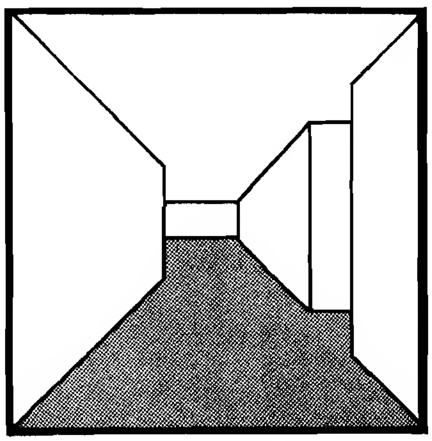
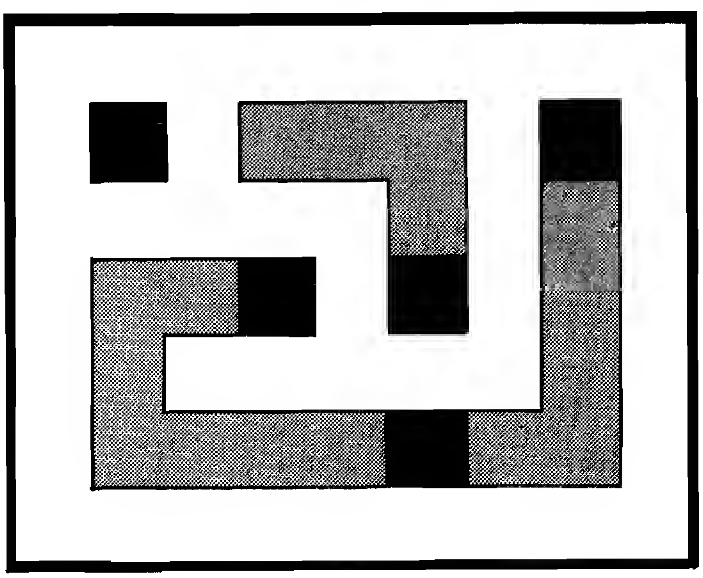


Figure 3



Listing 1

```
 Ø POKE 36879,14:POKE 45,47:POKE 46,22:POKE 55,196:POKE 56,22:CLR:
   PRINT"{PURPLE}STOP TAPE":GOTO 8
 1 I=-N*(M=1)-N*8*(M=2)-N*Y*(M=3):O=I+I:RETURN
 2 N=-1-(RND(8)> .5)*2:M=INT(RND(8)*3+1):RETURN
 3 GOTO 3ØØ
 4 G=255:POKE 84Ø,F AND G:POKE 842,R AND G:POKE 843,D AND G:
   POKE 841,-D AND G:POKE 844,-R AND G
 5 POKE 1,L/256:POKE Ø,L-PEEK(1)*256:PRINT"{CLEAR}":
   POKE 7713,PEEK(C+6634):POKE 36876,248:SYS 7448
 6 POKE 36878,.:POKE L,Z:GOTO 97
 8 WAIT 37151,64:A=7375:Y=48:B=6751:INPUT"LEVEL(Ø-9)";V:V=.Ø13*V+.3:
   J=RND(-TI):PRINT"WORKING;
 9 TI$="ØØØØØØ":L=68Ø8:FOR J=B TO A:POKE J,2:NEXT:FOR J=7327 TO A:
   POKE J,4:NEXT:FOR J=B TO 6799:POKE J,3:NEXT
3Ø FOR J=Ø TO 12:FOR K=B+J*Y TO B+8+J*Y:POKE K,3:NEXT:
   FOR K=B+J*Y TO 6791+J*Y STEP 8:POKE K,3:NEXT:NEXT:POKE L,1
41 GOSUB 2
42 X=X+1:GOSUB 1:IF RND(8)< V THEN 41
54 G=PEEK(L+I):IF PEEK(L+O)=2 AND G-3 THEN POKE L+I,1:L=L+O:POKE L,1:
   U=U+2:GOTO 42
55 IF G=4 THEN IF U> (V-.3)*288 THEN POKE L+I,5:
   PRINT U"STEPS TO EXIT":GOTO 66
58 IF G=1 AND INT(X/9)=X/9 THEN POKE L,2:POKE L+I,3:L=L+O:U=U-2:
   IF TI§B THEN U=Ø:GOTO 9
6Ø GOTO 41
66 FOR J=1 TO (V-.3)*677
68 L=B-1+INT(RND(8)*4+1)*2+INT(RND(8)*3+1)*16+INT(RND(8)*6+1)*96-56
7Ø IF PEEK(L)-1 THEN 68
72 FOR K=1 TO 6:GOSUB 1:IF RND(8)< .3 THEN 82
76 G=PEEK(L+I):
   IF PEEK(L+O)=2 THEN IF G-3 THEN IF G-5 THEN POKE L+I,1:L=L+O:
   POKE L,1:NEXT: GOTO 86
82 GOSUB 2:NEXT
86 NEXT:FOR J=B TO A:IF PEEK(J)> 2 AND PEEK(J)< 5 THEN POKE J,2
95 NEXT:F=1:R=-8:D=-Y:M=37151:N=M+1:X=M+3:E=1:O=197:B=B+8:L=68Ø8:
   PRINT"{CLEAR}":POKE 36869,255:GOTO 3
97 WAIT O,64:G=PEEK(M):IF G-94 AND G-126 THEN 97
98 POKE X,127:IF PEEK(N)-247 THEN 98
115 POKE X,127:IF PEEK(N)=119 THEN 2ØØ
12Ø POKE X,255:G=PEEK(M):IF G=122 THEN J=PEEK(L+F):
    IF J-2 AND J-5 THEN L=L+F:GOTO 3
125 IF G=118 THEN J=PEEK(L-F):IF J-2 AND J-5 THEN L=L-F:GOTO 3
13Ø J=R:IF G=11Ø THEN R=F:F=-J:GOTO 3
135 IF G=78 THEN R=-D:D=J:GOTO 3
14Ø J=F:IF G=86 THEN F=-D:D=J:GOTO 3
145 IF G=9Ø THEN F=D:D=-J:GOTO 3
15Ø G=PEEK(O):IF G=52 AND B> 6766 THEN B=B-8:GOTO 3
155 IF G=12 AND B< 6784 THEN B=B+8:GOTO 3
16Ø IF G=8 THEN E=-E:PRINT"{CLEAR}":GOTO 3
19Ø GOTO 115
2ØØ POKE X,255:IF PEEK(M)=94 THEN J=D:D=-R:R=J:GOTO 3
21Ø J=R:R=-F:F=J
3ØØ J=ABS(F):K=ABS(D):
    C=((1-(J> 1)-(J> 8))*SGN(F)+3)*7+(1-(K§1)-(K> 8))*SGN(-D)+3
31Ø POKE 834,PEEK(6634+C):Z=PEEK(L):POKE L,9:IF E-1 THEN 4
32Ø J=INT(B/256):POKE 1,B-J*256:POKE 2,J:SYS 7376:GOTO 6
```

Editor's Note: As a service to MICRO readers, the creators of this program will make it available on cassette.
Send $3.00 to:

Victory Software Corporation
1410 Russell Road
Paoli, PA 19301

FLOPPY DISKS SALE *$1.19 ea.

Economy Model or Cadillac Quality

LORAN CERTIFIED PERSONAL COMPUTER DISK ***We have the lowest prices!*** LORAN CERTIFIED PERSONAL COMPUTER DISK

***ECONOMY DISKS**

Good quality 5¼" single sided single density with hub rings.

Bulk Pac	100 Qty.	$1.19 ea.	Total Price	$119.00
	10 Qty.	1.39 ea.	Total Price	13.90

CADILLAC QUALITY (double density)

- ***Each disk certified*** • ***Free replacement lifetime warranty*** • ***Automatic dust remover***

For those who want cadillac quality we have the Loran Floppy Disk. Used by professionals because they can rely on Loran Disks to store important data and programs without fear of loss! Each Loran disk is 100% certified (an exclusive process) plus each disk carries an exclusive FREE REPLACEMENT LIFETIME WARRANTY. With Loran disks you can have the peace of mind without the frustration of program loss after hours spent in program development.

100% CERTIFICATION TEST

Some floppy disk manufacturers only sample test on a batch basis the disks they sell, and then claim they are certified. Each Loran disk is individually checked so you will never experience data or program loss during your lifetime!

FREE REPLACEMENT LIFETIME WARRANTY

We are so sure of Loran Disks that we give you a free replacement warranty against failure to perform due to faulty materials or workmanship for as long as you own your Loran disk.

AUTOMATIC DUST REMOVER

Just like a record needle, disk drive heads must travel hundreds of miles over disk surfaces. Unlike other floppy disks the Loran smooth surface finish saves disk drive head wear during the life of the disk. (A rough surface will grind your disk drive head like sandpaper). The lint free automatic CLEANING LINER makes sure the disk-killers (dust & dirt) are being constantly cleaned while the disk is being operated. PLUS the Loran Disk has the highest probability rate of any other disk in the industry for storing and retaining data without loss for the life of the disk.

Loran is definitely the Cadillac disk in the world

Just to prove it even further, we are offering these super LOW INTRODUCTORY PRICES

List $4.99 ea. INTRODUCTORY SALE PRICE $2.99 ea. (Box of 10 only) Total price $29.90

$3.33 ea. (3 quantity) Total price $9.99

All LORAN disks come with hub rings and sleeves in an attractive package.

DISK DRIVE CLEANER $19.95

Everyone needs a disk drive doctor

FACTS

- 60% of all drive downtime is directly related to poorly maintained drives.
- Drives should be cleaned each week regardless of use.
- Drives are sensitive to smoke, dust and all micro particles.
- Systematic operator performed maintenance is the best way of ensuring error free use of your computer system.

The Cheetah disk drive cleaner can be used with single or double sided 5¼" disk drives. The Cheetah is an easy to use fast method of maintaining efficient floppy diskette drive operation.

The Cheetah cleaner comes with 2 disks and is packed in a protective plastic folder to prevent contamination.

List $29.95 / Sale $19.95 * Coupon $16.95

Add $3.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!

VISA — MASTER CARD — C.O.D.

No C.O.D. to Canada, APO-FPO.

PROTECTO ENTERPRIZES (WE LOVE OUR CUSTOMERS)

BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

BIG FOUR

NEW 128K —MEGA BYTE DUAL DISK DRIVE—80 COLUMN

COMPUTER SYSTEM SALE!

HOME • BUSINESS • WORD PROCESSING

LOOK AT ALL YOU GET FOR ONLY **$895.**

	LIST PRICE
• B128 COMMODORE 128K 80 COLUMN COMPUTER	$ 995 00
• 8050 DUAL DISK DRIVE (over 1 million bytes)	1795.00
★ 4023 · 100 CPS · 80 COLUMN BIDIRECTIONAL PRINTER	499 00
• 12" HI RESOLUTION 80 COLUMN GREEN OR AMBER MONITOR	249 00
• BOX OF 10 LORAN LIFETIME GUARANTEED DISKS	49 95
• 1100 SHEETS FANFOLD PAPER	19 95
• ALL CABLES NEEDED FOR INTERFACING	102 05
TOTAL LIST PRICE	**$ 3717.95**

$895. SALE PRICE

Printer replacement options (replace the 4023 with the following at these sale prices)

	LIST	SALE
★ Olympia Executive Letter Quality Serial Printer	$ 699 00	**$ 399.00**
★ Comstar Hi-Speed 160 CPS 15½" Serial-Business Printer	$ 779 00	**$ 499.00**
★ Telecommunications Deluxe Modem Package	$ 199.00	**$ 139.00**

Plus You Can Order These Business Programs At Sale Prices

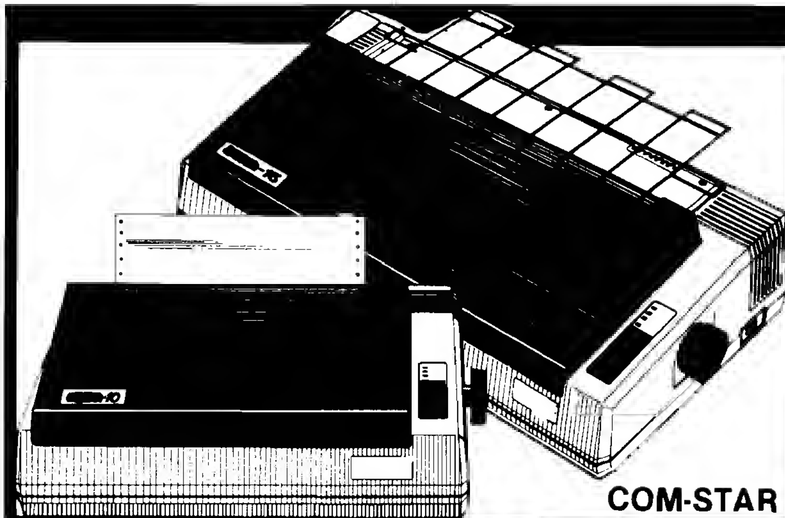
	LIST	SALE		LIST	SALE
Professional 80 Column Word Processor	$149 95	**$99.00**	Payroll	$149 95	**$99 00**
			Inventory	149 95	**99.00**
Professional Data Base	149 95	**99.00**	General Ledger	149 95	**99.00**
Accounts Receivable	149 95	**99.00**	Financial Spread Sheet	149 95	**99.00**
Accounts Payable	149 95	**99.00**	Program Generator	149 95	**99 00**

15 DAY FREE TRIAL We give you 15 days to try out this SUPER SYSTEM PACKAGE!! If it doesn't meet your expectations, just send it back to us prepaid and we will refund your purchase price!!
90 DAY IMMEDIATE REPLACEMENT WARRANTY If any of the SUPER SYSTEM PACKAGE equipment or programs fail due to faulty workmanship or material we will replace it IMMEDIATELY at no charge!!

Add $50.00 for shipping and handling!!
$100.00 for Canada, Puerto Rico, Hawaii orders.
WE DO NOT EXPORT TO OTHER COUNTRIES.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and MasterCard. We ship C.O.D. to U.S. addresses only.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

80 COLUMN PRINTER SALE—$149.00*



COM-STAR T/F

Tractor
Friction
Printer

only $ 179**

•15 Day Free Trial -180 Day Immediate Replacement Warranty

- Lowest Priced, Best Quality, Tractor-Friction Printers in the U.S.A.
- Fast 80-120-160 Characters Per Second
- 40, 46, 66, 80, 96, 132 Characters Per Line Spacing
- Word Processing
- Print Labels, Letters, Graphs and Tables
- List Your Programs
- Print Out Data from Modem Services
- "The Most Important Accessory for Your Computer"

*STX-80 COLUMN PRINTER—$149.00

Prints full 80 columns. Super silent operation, 60 CPS, prints Hi-resolution graphics and block graphics, expanded character set, exceptionally clear characters, fantastic print quality, uses inexpensive thermal paper! Best thermal printer in the U.S.A.! (Centronics Parallel Interface).

**DELUXE COMSTAR T/F 80 CPS PRINTER—$179.00

The COMSTAR T/F (Tractor Friction) PRINTER is exceptionally versatile. It prints 8½" x 11" standard size single sheet stationary or continuous feed computer paper. Bi-directional, impact dot matrix, 80 CPS, 224 characters. (Centronics Parallel Interface).

Premium Quality—120 CPS COMSTAR T/F SUPER-10X PRINTER—$289.00

COMSTAR T/F (Tractor Friction) SUPER-10X PRINTER gives you all the features of the COMSTAR T/F PRINTER plus a 10" carriage, 120 CPS, 9 x 9 dot matrix with double strike capability for 18 x 18 dot matrix (near letter quality), high resolution bit image (120 x 144 dot matrix), underlining, back spacing, left and right margin settings, true lower decenders with super and subscripts, prints standard, italic, block graphics and special characters, plus 2K of user definable characters! The COMSTAR T/F SUPER-10X PRINTER was Rated No. 1 by "Popular Science Magazine." It gives you print quality and features found on printers costing twice as much!! (Centronics Parallel Interface) (Better than Epson FX 80).

Premium Quality—120 CPS COMSTAR T/F SUPER-15½" PRINTER—$379.00

COMSTAR T/F SUPER 15½" PRINTER has all the features of the COMSTAR T/F SUPER-10X PRINTER plus a 15½" carriage and more powerful electronics components to handle large ledger business forms! (Better than Epson FX 100).

Superior Quality SUPER HIGH SPEED—160 CPS COMSTAR T/F 10" PRINTER—$399.00

SUPER HIGH SPEED COMSTAR T/F (Tractor Friction) PRINTER has all the features of the COMSTAR SUPER-10X PRINTER plus SUPER HIGH SPEED PRINTING—160 CPS, 100% duty cycle, 8K buffer, diverse character fonts, special symbols and true decenders, vertical and horizontal tabs. RED HOT BUSINESS PRINTER at an unbelievable low price!! (Serial or Centronics Parallel Interface)

Superior Quality SUPER HIGH SPEED—160 CPS COMSTAR T/F 15½" PRINTER—$529.00

SUPER HIGH SPEED COMSTAR T/F 15½" PRINTER has all the features of the SUPER HIGH SPEED COMSTAR T/F 10" PRINTER plus a 15½" carriage and more powerful electronics to handle larger ledger business forms! Exclusive bottom paper feed!!

PARALLEL INTERFACES

For VIC-20 and COM-64— $69.00
For Apple Computers— $79.00

NOTE: Other printer interfaces are available at computer stores!

Double Immediate Replacement Warranty

We have doubled the normal 90 day warranty to 180 days. Therefore if your printer fails within "180 days" from the date of purchase you simply send your printer to us via United Parcel Service, prepaid. We will IMMEDIATELY send you a replacement printer at no charge, prepaid. This warranty, once again, proves that WE LOVE OUR CUSTOMERS!

Add $17.50 for shipping, handling and insurance. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. VISA — MASTER CARD ACCEPTED. We ship C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

SUPER-10" ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890

• Graphic Print for C-64 •

Part 2

by Michael J. Keryan

Create full-page graphic printouts interfacing with major commercial software.

Last month we saw how to add a fast machine language routine to dump a graphic screen to most popular non-Commodore printers. This month we add another machine language program and a BASIC program that can load graphic files from a number of popular graphic programs, display them, and dump them to a printer.

Last time, a general purpose graphic dump program was described (GDUMP). This program will give HiRes or MULTIcolor graphic dumps in various dot patterns — the density of the dot pattern is proportional to the darkness of the actual colors used in the picture. Sixteen different patterns are used so that even two colors that look identical on a black and white monitor or TV can be distinguished on the printout. The printouts are about 7x9 inches and take from two to two and a half minutes, depending on your printer speed. Printers supported are NEC 8023, PROWRITER, C. ITOH 8510, EPSON MX-80, FX-80, GEMINI, and any other printer that emulates graphic modes of these printers.

Now we have a program that will print graphics. But what are we going to print? Well, it would be nice if we could use this program to print all our neat pictures, graphs, etc. that we developed with those graphic aid and drawing programs that we paid anywhere from $20 to $75 for. You remember, the ones that didn't come with a screen dump program or the ones that took three minutes to produce a picture small enough to stick in your wallet? To print these, we'll first have to transform their graphs to memory areas we can easily get to. We'll need a machine language program to do this because we'll want to move between 8K and 10K of memory. But first we need a BASIC program that makes things a lot simpler.

A small BASIC program (see Listing 1) ties everything together. The first thing this program does is load the machine language programs GDUMP (from last time) and GMOVE (Listing 2). These were combined into one program "GDUMP+MOVE". Then a jump is made to line 2000 which changes the screen colors and displays a menu. You are given the choice of dumping graphics made from SIMONS' BASIC, ULTRABASIC-64, DOODLE, KOALAPAINTER, or SLIDESHOW. A sixth option allows SLIDESHOW graphics to be inverted.

If the picture is stored on a disk (options 2-6), you are instructed to put in the disk containing the picture and type in the name of the file. For DOODLE files, include the DD as part of the file name as listed on the directory. For KOALA files, include the PIC part, such as "PIC H CASTLE", but don't worry about the initial reverse field character that shows up in the directory. All file names can be shortened, but don't use the * wildcard; for example "PIC H" will work fine.

The picture is loaded into the same memory areas that these other programs use. Our BASIC program then jumps to line 1050. The graphic screen is reconfigured by a SYS to GMOVE. The workings of GMOVE is described later. Printer specific controls are set up (lines 1050-1100), then the keyboard is polled. If you hit P, you will get a printout. Any other key will reset the screen and end the program with no printed output.

SIMONS' BASIC

SIMONS' BASIC is a new package of BASIC extensions, distributed by Commodore. It comes in a plug-in cartridge form and reduces the size of BASIC workspace by 8K bytes. With SIMONS' BASIC, you get 114 more BASIC commands in 12 general categories. This is a super package and is a steal at $20. A drawback is that it uses the $CXXX area of RAM, making most of your machine language programs incompatible. However, this package contains so much utility that you may not need other ML programs.

With SIMONS' BASIC, you can make pictures, graphs, etc. in HiRes or MULTIcolor modes. You are given quite a few graphic commands for drawing lines, circles, blocks, adding text, etc. It has a COPY command that will dump the graphic screen to a printer in Commodore 1525 format.

Although the copy command is handy, it has a few undesirable features. It is small (4 x 2 3/4 on my printer) and slow (3 minutes). SIMON'S BASIC allows color redefinition, but only 3 colors can be specified at any time. With COPY, colors 1 and 2 come out as half-tone (vertical lines), while color 3 comes out solid black. Your highly colored pictures may look great on the screen, but the printouts may lack a little contrast between colors.

SIMONS' BASIC places the 8K bit map memory in hidden RAM, under the KERNAL at $E000-$FFFF. The 1K screen memory is placed at $C000 (normally found at $0400). The routine GDSIMN (see Listing 2) switches out the ROM, allowing you to access the 8K bit map data, transfers the memory to $2000-$3FFF, and then switches the ROM back in. It also places $C0 into the screen pointer. Since SIMONS' BASIC contains no command to dump a graphic screen to disk, you will have to print the screen while it is displayed. By running the program in Listing 1, you can create a 7 line BASIC program for SIMONS' BASIC. Append your graphic program to this one with the SIMONS' BASIC MERGE command. Then when you want the graphic screen dumped to the printer, just include the line GOTO 1. This short program uses only one variable, A. Make sure you don't use A in the body of your program

ULTRABASIC-64

ULTRABASIC-64 is a package of 50 extra BASIC commands, made by Abacus Software. The graphic

commands are quite similar to those of SIMONS' BASIC. A graphic screen dump using the HARD command is also similar: it is fairly slow, small, and docs not give accurate color shading renditions. With ULTRABASIC-64 dumps, you get 3 different shade densities for colors defined as 1, 2, and 3. But the darkness is a function of the color number (1, 2, 3) which is not related to the actual darkness of that color.

With ULTRABASIC-64, you can save a graphic screen to disk with the DUMP command or by pressing function key F2. This file can later be read in, reconfigured, and printed with all the attributes of GDUMP. GDULTR in Listing 2 first switches out the ROMS, moves the 8K bit map area of memory located at $A000-$BFFF to $2000-$3FFF, then re-enables the ROMS. The program then relocates the 1K screen memory starting at $8400 to $0400, and the 1K color memory starting at $8800 to $D800. The border color is moved from $83E0 to $D020, and the background color from $83E1 to $D021. The screen is then configured for bit-mapped graphics and $83D6 is transferred to $D016 to enable either HiRes or MULTI modes.

DOODLE

DOODLE is a very extensive drawing program, by Omni Unlimited. Although somewhat difficult to learn due to the many options and menus, you can create quite outstanding drawings with a joystick. Because DOODLE uses the HiRes mode, you can get lines that are only one dot wide; most other color drawing packages allow only two-dot resolution due to the use of MULTI mode. In any 8x8 square of dots, you can display any two colors: one for the dots and another for the background. However, the colors in any other 8x8 square block can be completely different. Of course, DOODLE handles all this for you; you just draw the pictures.

DOODLE provides a printer setup program for non-Commodore printers and a graphic dump program which is better than most other graphic packages attempts. It allows two sizes and is fast. The printer dumps have one fault, however. All dots are printed as black and all background is printed as white — no matter what colors were used for the dots and background. Check out the Middle Earth demo that is on thc DOODLE disk. The white clouds against the blue sky are printed as black clouds on a white sky.

DOODLE contains a routine to save a picture to disk. It creates a file with DD as the first two characters of the file name. With the programs provided in Listings 1 and 2, you can read in the file, reconfigure and display it, and print it with GDUMP. GDDOOD in Listing 2 moves the 8K bit map area from $6000-$7FFF to $2000-$3FFF, and the 1K screen area from $5C00-$5FFF to $0400-$07FF. Then the bit-mapped graphic mode is enabled and the HiRes mode is enabled, displaying the picture.

KOALAPAINTER

KOALAPAINTER is a software package that you get on disk when you purchase a KOALA PAD for the Commodore 64. With this program (by Audio Light/Koala Technologies), it's comprehensive single-page menu, and the KOALA PAD, even a six year old can quickly learn to make quite attractive graphics. This program is super user-friendly while being quite powerful.

At this time, KOALA provides no routine to dump your pictures to a printer, but will probably provide one in the future (for a price). KOALA does provide a routine to save your pictures to a disk file. This file can be used with the programs here to get a printer dump. In Listing 2, GDKOAL moves the 8K bit map area located at $6000-$7FFF to our common area of $2000-$3FFF. The 1K screen area starting at $7F40 is moved to $0400, and the 1K color memory starting at $8328 is moved to $D800. The background color is moved from $8710 to $D021. Then the bit mapped screen in turned on and the MULTIcolor mode is enabled, displaying the KOALA-produced picture.

SLIDESHOW

SLIDESHOW is a program that has appeared on several TPUG (Toronto Pet Users Group) disks. It uses a machine language program called HRSUPP to clear color memory and display a high resolution bit-mapped graphic picture loaded from disk into $2000-$3FFF. Quite a few digitized pictures are available in this format,

including a photo of Ronald Reagan. SLIDESHOW uses the HiRes mode using only two colors; some pictures are shown in light on dark, others as dark on a light background.

SLIDESHOW pictures can be viewed and printed with the programs provided here. GDSLID in Listing 2 first clears the screen memory (1K block at $0400) by creating black dots on a white background. Then the bit-mapped mode is enabled as is the HiRes mode, allowing the pictures to be displayed.

PRINTER SET-UP

Before running Listing 1, first make sure the printer setup matches your printer and interface. Four constants are incorporated in the program (lines 2120-2250). PT is the printer type: 0 for C. Itoh 8510, Prowriter, and NEC 8023, and 1 for Epson MX80 with GRAFTRAX or FX80 or compatible printers like Gemini/Star. NP is the repeat counter and is automatically set up from PT. NT is the interface type: 0 for Tymac Connection and 1 for others. SD is the secondary address required by your interface for transparent operation (use 6 for Connection, 5 for Cardco). After these changes are made, save the program to disk.

To save you the effort of typing in these programs, they are being made available as a MicroDisk. The MicroDisk, number MD-4, contains all programs in this series. In the last installment, we will get into what you have been waiting for — a method to print your pictures in full color on your existing printer.

Listing 1

```
                    ;SUPPORT PROGRAM FOR GDUMP
                    ;     M.J.KERYAN  3-25-84
                    ;
5530Ø                        ORG $553Ø
                    ;
553Ø 4C 3F 55       GOSIMN   JMP GDSIMN
                    ;FOR SIMON'S BASIC SCREEN DUMP
                    ;  A TRADEMARK OF COMMODORE ELECTRONICS
5533 4C 8F 55       GOULTR   JMP GDULTR
                    ;FOR ULTRABASIC-64 SCREEN DUMP
                    ;  A TRADEMARK OF ABACUS SOFTWARE
5536 4C D1 55       GODOOD   JMP GDDOOD
                    ;FOR **DOODLE** SCREEN DUMP
                    ;  A TRADEMARK OF OMNI UNLIMITED
5539 4C F6 55       GOKOAL   JMP GDKOAL
                    ;FOR KOALAPAINTER SCREEN DUMP
                    ;  A TRADEMARK OF AUDIO LIGHT
553C 4C 38 56       GOSLID   JMP GDSLID
                    ;FOR SCREEN DUMP OF PICTURES IN
                    ;  SLIDESHOW FORMAT (TPUG)
                    ;
553F 2Ø 4E 55       GDSIMN   JSR ROMOUT      ;SWITCH OUT ROMS
5542 2Ø 5A 55                JSR GD1         ;MOVE MEMORY
5545 2Ø 54 55                JSR ROMIN       ;ROMS BACK IN
5548 A9 CØ                   LDA #$CØ
554A 8D Ø9 5Ø                STA $5ØØ9
554D 6Ø                      RTS
                    ;
554E 78             ROMOUT   SEI             ;IGNORE INTER.
554F A9 34                   LDA #$34        ;ROMS OUT
5551 85 Ø1                   STA $Ø1
5553 6Ø                      RTS
                    ;
5554 A9 37          ROMIN    LDA #$37        ;ROMS BACK IN
5556 85 Ø1                   STA $Ø1
5558 58                      CLI             ;RESTORE INTER.
5559 6Ø                      RTS
                    ;
555A A9 EØ          GD1      LDA #$EØ        ;SET UP
555C 8D 7Ø 55       GD2      STA FROM+2
555F A9 2Ø          GD3      LDA #$2Ø
5561 8D 73 55                STA TO+2
5564 AØ 2Ø                   LDY #$2Ø        ;32 BLOCKS
5566 A2 ØØ          GD4      LDX #$ØØ
5568 8E 6F 55                STX FROM+1
556B 8E 72 55                STX TO+1
556E BD ØØ EØ       FROM     LDA $EØØØ,X     ;MOVE DATA
5571 9D ØØ 2Ø       TO       STA $2ØØØ,X
5574 E8                      INX
5575 DØ F7                   BNE FROM
```

```
5577 EE 7Ø 55                INC FROM+2
557A EE 73 55                INC TO+2
557D 88                      DEY
557E DØ EE                   BNE FROM
558Ø 6Ø                      RTS
5581 A9 84          GD5      LDA #$84        ;SET UP
5583 8D 7Ø 55       GD6      STA FROM+2
5586 A9 Ø4          GD7      LDA #$Ø4
5588 8D 73 55                STA TO+2
558B AØ Ø4          GD8      LDY #$Ø4        ;4 BLOCKS
558D DØ D7                   BNE GD4         ;BRANCH ALWAYS
                    ;
558F 2Ø 4E 55       GDULTR   JSR ROMOUT      ;ROMS OUT
5592 A9 AØ                   LDA #$AØ
5594 2Ø 5C 55                JSR GD2
5597 2Ø 54 55                JSR ROMIN       ;ROMS BACK IN
559A 2Ø 81 55                JSR GD5
559D A9 88                   LDA #$88
559F 8D 7Ø 55                STA FROM+2
55A2 A9 D8                   LDA #$D8
55A4 8D 73 55                STA TO+2
55A7 2Ø 8B 55                JSR GD8
55AA AD EØ 83                LDA $83EØ       ;BORDER COLOR
55AD 8D 2Ø DØ                STA $DØ2Ø
55BØ AD E1 83                LDA $83E1       ;BACKGROUND
55B3 8D 21 DØ                STA $DØ21
55B6 2Ø CØ 55                JSR BITON       ;BIT-MAP ON
55B9 AD D6 83                LDA $83D6
55BC 8D 16 DØ                STA $DØ16       ;HIRES OR M-C
55BF 6Ø                      RTS
                    ;
55CØ AD 11 DØ       BITON    LDA $DØ11
55C3 Ø9 3Ø                   ORA #$3Ø
55C5 8D 11 DØ                STA $DØ11       ;BIT MAP ON
55C8 AD 18 DØ                LDA $DØ18
55CB Ø9 Ø8                   ORA #$Ø8        ;LOCATED AT
55CD 8D 18 DØ                STA $DØ18       ; $2ØØØ
55DØ 6Ø                      RTS
                    ;
55D1 A9 6Ø          GDDOOD   LDA #$6Ø
55D3 2Ø 5C 55                JSR GD2         ;MOVE BIT-MAP
55D6 A9 5C                   LDA #$5C
55D8 2Ø 83 55                JSR GD6         ;MOVE SCREEN
55DB 2Ø CØ 55                JSR BITON       ;BIT MAP ON
55DE 2Ø E2 55                JSR MCOFF       ;MULTI COLOR OFF
55E1 6Ø                      RTS
                    ;
55E2 AD 16 DØ       MCOFF    LDA $DØ16       ;TURN OFF
55E5 29 CF                   AND #$CF        ;MULTI-COLOR
55E7 8D 16 DØ                STA $DØ16       ;MODE
55EA 6Ø                      RTS
                    ;
```

```
55EB AD 16 D0   MCON     LDA $D016      ;TURN ON
55EE 29 DF               AND #$DF       ;MULTI-COLOR
55F0 09 10               ORA #$10       ;MODE
55F2 8D 16 D0            STA $D016
55F5 60                  RTS
                ;
55F6 A9 60      GDKOAL   LDA #$60       ;MOVE BIT-MAP
55F8 20 5C 55            JSR GD2
55FB A2 00               LDX #$00
55FD 8E 72 55            STX TO+1
5600 A0 04               LDY #$04
5602 8C 73 55            STY TO+2
5605 A9 40               LDA #$40
5607 8D 6F 55            STA FROM+1
560A A9 7F               LDA #$7F
560C 8D 70 55            STA FROM+2
560F 20 6E 55            JSR FROM       ;MOVE SCREEN
5612 A2 00               LDX #$00
5614 8E 72 55            STX TO+1
5617 A0 04               LDY #$04
5619 A9 D8               LDA #$D8
561B 8D 73 55            STA TO+2
561E A9 28               LDA #$28
5620 8D 6F 55            STA FROM+1
5623 A9 83               LDA #$83
5625 8D 70 55            STA FROM+2
5628 20 6E 55            JSR FROM       ;MOVE COLOR MEM
562B AD 10 87            LDA $8710
562E 8D 21 D0            STA $D021      ;BACKGROUND
5631 20 C0 55            JSR BITON      ;BIT-MAP ON
5634 20 EB 55            JSR MCON       ;MULTI-COLOR ON
5637 60                  RTS
                ;
5638 A0 04      GDSLID   LDY #$04
563A 8C 46 56            STY GSL+2
563D A2 00               LDX #$00
563F 8E 45 56            STX GSL+1
5642 A9 10               LDA #$10       ;SCREEN COLORS
5644 9D 00 04   GSL      STA $0400,X    ;SET TO BLACK
5647 E8                  INX            ;AND WHITE
5648 D0 FA               BNE GSL
564A EE 46 56            INC GSL+2
564D 88                  DEY
564E D0 F4               BNE GSL
5650 20 C0 55            JSR BITON      ;BIT-MAP ON
5653 20 E2 55            JSR MCOFF      ;MULTI COLOR OFF
5656 60                  RTS
5657                     END
```

Listing 2

```
1000 REM BASIC PROGRAM TO SUPPORT GDUMP
1010 REM      M.J.KERYAN  3-30-84
1020 :
1030 IF A=0 THEN A=1: LOAD" GDUMP+MOVE",8,1
1040 IF A=1 THEN A=2: GOTO 2000
1050 POKE 20491,PT: POKE 20492,SD
1060 POKE 20493,NT: POKE 20487,NP
1070 SYS GT
1080 IF TY=2 OR TY=4 THEN MD=PEEK(53270):
     MD=3-((MD AND 16)/16): POKE 20494,MD
1090 IF TY=3 OR TY=5 THEN POKE 20494,3
1100 IF TY=6 THEN POKE 20494,0
1110 GETK$:IF K$<>""THEN 1110
1120 GETK$:IF K$="" THEN 1120
1130 IF K$="P" THEN SYS 20480
1140 POKE 53265,(PEEK(53265)AND223)
1150 POKE 53270,(PEEK(53270)AND207)
1160 POKE 53272,21
1170 POKE 53280,6: POKE 53281,15: POKE 646,0
1180 PRINT"{CLEAR}": END
2000 POKE 53280,6: POKE 53281,15: POKE 646,0
2010 PRINT"{CLEAR,DOWN2}WHICH TYPE OF PICTURE?"
2020 PRINT
2030 PRINT"    1  SIMON'S BASIC"
2040 PRINT"    2  ULTRABASIC-64"
2050 PRINT"    3  DOODLE"
2060 PRINT"    4  KOALAPAINTER"
2070 PRINT"    5  SLIDESHOW"
2080 PRINT"    6  SLIDESHOW - INVERTED"
2090 INPUT"  ";TY
2100 IF TY<1 OR TY>6 THEN 2000
2110 :
2120 PT = 0:  REM PRINTER TYPE
2130 :        REM  NEC/PROWRITER    = 0
2140 :        REM  EPSON OR SIMILAR = 1
2150 :
2160 NP = 3:  IF PT=1 THEN NP = 2
2170 :        REM REPEAT   CODE
2180 :
2190 NT = 0:  REM INTERFACE TYPE
2200 :        REM  CONNECTION = 0
2210 :        REM  OTHERS     = 1
2220 :
2230 SD = 6:  REM SECONDARY ADDRESS
2240 :        REM FOR TRANSPARENT
2250 :
2260 GT = 21808 + (TY-1)*3
2270 IF GT>21820 THEN GT=21820
2280 IF TY=1 THEN 3000
2290 PRINT"{DOWN2}
     NOW PUT IN DISK WITH THE PICTURE FILE."
2300 INPUT"{DOWN}NAME OF PICTURE";NM$
2310 PRINT"{DOWN}AFTER PICTURE LOADS, PRESS:"
2320 PRINT"   P TO PRINT IT"
2330 PRINT"   E TO EXIT"
2340 IF TY=4  THEN LOAD "?"+NM$+"*",8,1
2350 IF TY<>4 THEN LOAD NM$+"*",8,1
2900 :
3000 REM CREATE A SIMON'S BASIC PROGRAM
3010 Q$=CHR$(34)
3020 PRINT"{CLEAR}1 IF A=1 THEN A=2:
     LOAD"Q$" GDUMP+MOVE"Q$",8,1"
3030 PRINT"2 IF A=0 THEN A=1: GOTO 7
3040 PRINT"3 POKE 20491,"PT" :POKE 20492,"SD
3050 PRINT"4 POKE 20493,"NT" :POKE 20487,"NP" :
     SYS 21808"
3060 PRINT"5 A=PEEK(53270): A=(A AND 16)/16"
3070 PRINT"6 A=3-A: POKE 20494,A: SYS 20480: END"
3080 PRINT"7 REM APPEND YOUR PROGRAM HERE"
3090 PRINT"SAVE"Q$"SIMON.GDUMP"Q$",8"
3100 POKE 631,19: FOR A=632 TO 639: POKE A,13: NEXT A
3110 POKE 198,9: NEW
```

* NOTE: Program **GDUMP** from last month (MICRO 73:22) should be combined with above program **GMOVE** into a new program **GDUMP + GMOVE**.

feature

Alter Track & Sector on Vic-20 & C-64

by Edwin L. King

Rewrite any sector on a disk without any loss of data.

Requirements: VIC-20, C64 or any model PET with disk drive.

The ability to examine and modify information on the disk is rather like the plumber's flaring tool. One does not need it often, but there is no such thing as a substitute. As a high school student, I was allotted one disk. After about six months of work, including developing a very fine adventure game that was 39 blocks long, the disk header got 'confused'. These was no way to use the disk without the header. The demo program DISPLAY T&S showed that at least some of the data was still good. Two things happened: first, I NEWed the disk, lost the data, and was never able to successfully recreate it; second, I developed this utility.

The program is an odd sort of hybrid. It was written and debugged on a PET 4032, revised on a VIC-20 so that it definitely works on the 4040/2031/1540 disk systems and should work on the 8050 as well, and finally was tested on the C64. It is straightforward and requires little, if any, external explanation. The BLOCK-READ/WRITE commands are used quite frequently. After all of the preliminary data is INPUTed from the user, the requested sector is displayed in a hex memory dump of the format:

Listing 1

```
1 REM ##################################
2 REM ## ALTER T&S FOR 4040 DISK AND ##
3 REM ## PET 4032        BY THE FUZZ ##
4 REM ##################################
5 REM
6 REM     ++++SET CONSTANTS++++
7 REM
10 T=0:REM  CURRENT TRACK
11 S=0:REM  CURRENT SECTOR
12 D=0:REM  DRIVE
15 HX$="0123456789ABCDEF":REM HEXCON
16 REM +++ALL OTHERS ARE TEMPORARY+++
17 DIM P$(42),R$(255)
18 OPEN 15,8,15:OPEN 2,8,2,"#0":GOSUB 10000
20 REM
21 REM   ++++INPUT FROM USER++++
22 REM
25 PRINT"{CLEAR,RVS} ALTER T&S BY FUZZ "
26 PRINT:PRINT:PRINT
27 INPUT"DRIVE";D
28 INPUT"{DOWN3}TRACK,SECTOR";T,S

29 IF T=0 OR T>35 THEN PRINT#15,"I0":CLOSE 15:
   CLOSE 2:PRINT"{CLEAR,RVSOFF,RVS}END":END
30 GOSUB 500: REM READ T&S INTO R$(I)
40 GOSUB 600: REM FOR P$'S
45 PRINT"{CLEAR}TRACK:"T"    SECTOR:"S:
   PRINT:PRINT:PRINT
46 PRINT"OUTPUT TO {RVS}S{RVSOFF}
   CREEN   OR {RVS}P{RVSOFF}RINTER"
47 GET QW$:IF QW$<>"S" AND QW$<>"P" THEN 47
48 IF QW$="P" THEN OPEN 3,4:DQ=4:GOTO 50
49 OPEN 3,3:DQ=3
50 FOR I=1 TO K-1:PRINT#3,P$(I)
51 IF I<>INT(K/2) OR DQ=4 THEN 55
52 PRINT"{DOWN2}PRESS ANY KEY TO CONTINUE"
53 GET QW$:IF QW$="" THEN 53
55 NEXT I
```

```
00: 01 23 3F 4B CC 02 0A AB
08: 20 30 40 50 60 70 80 90
        .
        .
        .
F8: 01 02 03 04 05 06 70 80
```

The index-looking things are just that: "line numbers" your way of indicating which line(s) you wish to change in that sector. Naturally, the program can change any block on the disk, fix spelling in files or the directory, all under your watchful eye and direct control.

Disk Headers

Creation of a new disk header requires knowledge of how data is stored on the disk. I would suggest a quick review of the section on advanced disk programming in your manual before making any modifications to the disk. The first thing that must be done is to make sure the disk has some of its house keeping in order. To do this, follow the program prompts to modify these lines to read as follows:

```
00: 12 01 41 00 $$ $$ $$ $$

A0: $$ $$ $$ $$ A0 32 41 $$
```

Do **not** type in the dollar signs! They are there to remind you that there will be data in those locations that should not be disturbed.

Next you must give the disk an **ID** number. Select a two digit number. Convert it into its hex ASCII value by placing the digit '3' in front of each of the digits. Enter these two values in the third and fourth positions on line A0: at locations A2 and A3. For example, if the ID number choosen was 19, then the two hex ASCII digits would be '31' and '39', and the line would look like:

```
A0: $$ $$ 31 39 A0 32 41 $$

           1  9
```

The disk must have a name. This is accomplished by converting each of the letters in the selected disk name into their hex ASCII values and entering them into locations 90 through A1. If the name is less than eighteen (18) characters long, then pad the remaining locations with the shifted space character, hex A0. For example, if the disk were to be named **GAMES**, then the lines modified so far would look like:

```
56 IF DQ=4 THEN PRINT#3:CLOSE 3
60 PRINT"ARE THERE ANY MODIFICATIONS":INPUT M$
61 IF LEFT$(M$,1)="N" THEN 90
70 PRINT:PRINT"WHICH LINE";:INPUT L$
72 FOR I=1 TO K:IF LEFT$(P$(I),2)=L$ THEN Q=I:I=1000
73 NEXT
74 IF I< K+1 THEN 60
80 PRINT"{RIGHT}"RIGHT$(P$(Q),LEN(P$(Q))-3);:
   PRINT"{LEFT26}";
81 INPUT Q$:P$(Q)=LEFT$(P$(Q),3)+" "+Q$
82 GOTO 60
90 GOSUB 200:REM      BREAK P$'S
95 GOSUB 100:REM      UPDATE T&S
97 PRINT"{CLEAR}":GOTO 28
100 REM +++++++++WRITE TO DISK++++++++++
110 PRINT#15,"B-P"2;1
130 PRINT"{CLEAR}":FOR I=1 TO 255:PRINT#2,R$(I);:
    PRINT"{HOME}BYTE"I" OF 255":NEXT
140 PRINT#15,"U2:"2;D;T;S:RETURN
200 REM ++++++UPDATE R$(I)++++++
210 FOR I=1 TO K-1:P$(I)=RIGHT$(P$(I),LEN(P$(I))-4):
    NEXT:M=0
211 PRINT"{CLEAR}PLEASE HOLD WHILE I UPDATE THE DISK"
220 FOR I=1 TO K-1
230 PRINT"{CLEAR}":FOR J=1 TO LEN(P$(I))STEP3
240 X$=MID$(P$(I),J,2):GOSUB710
250 R$(M)=CHR$(X):M=M+1:PRINT"{HOME}BYTE"M"="X$
260 NEXTJ:NEXTI
270 RETURN
500 REM  ++++READ FROM DISK++++
510 PRINT#15,"U1:"2;D;T;S
515 GOSUB10000
520 PRINT#15,"B-P:"2;1
521 PRINT#15,"M-R"CHR$(0)CHR$(17)
522 GET#15,R$(0)
530 FOR I=1 TO 255:GET#2,R$(I):NEXT
540 PRINT#15,"B-P:"2;1
550 RETURN
600 REM ++++FORM PRINT STRINGS++++
605 PRINT" PLEASE HOLD WHILE I   ARRANGE THE DATA"
610 K=1:FOR I=0 TO 255
615 IF R$(I)="" THEN P=0:GOTO 630
620 P=ASC(R$(I))
630 X=P:GOSUB 800
640 P$(K)=P$(K)+X$+" "
650 IF (I+1)/8=INT((I+1)/8) THEN K=K+1
660 NEXT
670 FOR I=0 TO K-1
680 X=I*8:GOSUB 800
690 P$(I+1)=X$+":  "+P$(I+1)
700 NEXT:RETURN
710 REM +++++++HEX TO DEC+++++++++++
720 L$=LEFT$(X$,1):H$=RIGHT$(X$,1)
730 X=0:FOR B=0 TO 15
740 IF L$<> "" THEN IF MID$(HX$,B+1,1)=L$
    THEN X=X+16*B:L$=""
750 IF H$<> "" THEN IF MID$(HX$,B+1,1)=H$
    THEN X=X+B:H$=""
760 NEXT B
770 RETURN
800 REM +++++DEC TO HEX+++++
810 H=INT(X/16):L=INT(X-H*16)
820 H=H+1:L=L+1
830 X$=MID$(HX$,H,1)+MID$(HX$,L,1)
840 RETURN
10000 REM +++++ GET ERROR STATUS +++++
10010 INPUT#15,EN,EM$,ET,ES
10020 IF EN THEN PRINT#15,"I0":CLOSE15:
      CLOSE2:PRINT"{CLEAR,RIGHT9,DOWN10}"EM$:STOP
10030 RETURN
```

```
ØØ: 12 Ø1 41 ØØ $$ $$ $$ $$

9Ø: 47 41 4D 45 52 AØ AØ AØ

    G  A  M  E  S

98: AØ AØ AØ AØ AØ AØ AØ AØ

AØ: AØ AØ 31 39 AØ 32 41 $$
```

Now comes the fun part — outsmarting the computer! The only thing left to do is to rewrite the **BAM** (Block Availability Map), a special type of map in the header in which one bit equals one sector and a one in that bit indicates that the sector should not be overwritten. If there is enough data on the disk being fixed to make using this program better than **NEW**ing the disk, this would seem like an impossible, or at least forbiddingly tedious, task. But, as usual, there is an easier way. First, exit the program by answering **NO** to the question about modifications and **0,0** when asked for the track and sector. This will cause the program to update track 18, sector 0, which the is disk header that we have been working on. Once this is done, execute in immediate mode a **COLLECT** command (on the 1540 that is a **VALIDATE**.) This forces the disk drive to update its own BAM, freeing you from the task. If the header was the only thing damaged, this may be enough to correct the problem. Otherwise, it may be necessary to play with a few of the directory tracks or program/data sectors. I leave that to the ambitious, experienced or desparate! I would suggest that, as soon as this is done or when the disk is working again, that you back it up **IMMEDIATELY** (as in right now and without hesitation).

Program Description

The most tragically ironic thing that can happen to a CBM floppy disk is the loss of the header, particularly when the information and directory remain intact. The program is quite straightforward. Lines 10 to 15 list the permanent, or relatively permanent, variables such as current track, sector and drive. Line 17 DIMensions the two arrays: R$(I) holds the single byte characters exactly as they are received from the disk and P$(I) holds the line index and 8 bytes for printing, inputting and other tasks.

Lines 25 through 97 handle all input from the user. Once the requested track and sector are determined, subroutine 500 loads the 256 bytes from the requested disk block into the buffer and then into the R$(I) array. Subroutine 600 converts characters into their ASCII hex form, adds the line index to generate the output strings in array P$(I). This array is dumped to the printer or display in lines 45 to 56. Lines 60 through 82 handle changes to the block. Subroutine 200 converts the updated P$(I) array back to the R$(I) array. The routine at line 100 rewrites the modified disk sector. Since these last two routines are a little time consuming, a few print statements are included to break the boredom. The complete run requires about 3 to 5 minutes per sector. Exit from the program occurs when an illegal track is requested: less than 1; greater than 35. **WARNING**: Under no circumstances should you tamper with line 120. The result could be a one-way trip to Disk-Never-Never-Land.

Edwin is currently attending Florida State University, majoring in Computer Science. He was introduced to computers at the Governor's Honors Program in Georgia in 1981 and followed up this start with a Computer Math class in high school where he learned BASIC and misc. PET stuff.

The UCSD p-System: De Facto Standard 68000 Operating System

by Paul Lamar & Richard Finder

An explanation of the p-System and a comparison of six 68000-based computer systems.

It may be a result of reading an overabundance of IBM PC ads that makes people, without knowledge of microprocessor architecture or assembly language, blatantly predict that MS-DOS on the eight bit 8088 chip will become the measure by which all operating systems and microcomputers will be judged during the coming decade. That view is simply wrong and such comments (especially by people who should know better) may be the result of an understandable impatience with the performance of slow, memory limited, eight bit microcomputers — but to declare that MS-DOS and the 80XXX is going to be the de facto industry standard is short-sighted at best and misleading at worst. Ironically, the IBM PC fits the slow 8 bit category. There are other operating systems and microprocessors out there, more powerful than MS-DOS and the 8088. One such is the p-System running on the 68000.

The p-System is a large, hierarchical menu driven portable operating system that is available on many different mini and microcomputers. There is no need to remember cryptic commands as in most other operating systems. All commands are shown in English, on a menu line, at the top of the screen.

A group of students at the University of California at San Diego began writing the p-System under the direction of Dr. Kenneth L. Bowles in 1974. A minicomputer with a hard disk was used as a "development system," a procedure whose significance will become apparent later.

Dr. Bowles originally called it "the UCSD PASCAL SYSTEM," which was a tactical error; not everybody likes the PASCAL language so some people were "put off" by this aspect of the original operating system. The present P-System has nothing to do with any specific High Level Language (HLL), many HLLs other than PASCAL run under the p-System including BASIC. It is now a stand alone operating system. The p-System was written in PASCAL and then compiled to P-Code. To give you an idea of the significance of this statement, I will try to explain a few facts about compilers and interpreters. (A compiler is a utility program that

converts HLL programs to native machine codes in one continuous operation, before any attempt is made to run the HLL program.)

P-Code is a compact intermediate code that is interpreted by a P-Code interpreter program. HLL programs written with interpreters run much slower than the same programs written with a compiler because this conversion has to be done while the HLL program is running. Compilers do it only once; thereafter, only the native machine code is run. Here, the interpreter is converting a partially compiled, intermediate P-Code instead of HLL source code. This scheme results in program code that is compact and that executes faster than a pure interpreter, but not as fast as a pure compiler.

When implementing the p-System, a P-Code interpreter program is written for each different type of microprocessor, it being quicker and cheaper to write a P-Code interpreter than it is to write a complete microprocessor specific compiler. This allowed the p-System to be quickly transported to new microprocessors as they appear. That was important ten years ago. However, we can expect that fewer entirely new, general purpose microprocessors will be introduced in the foreseeable future.

General purpose microprocessors are becoming so complex and require so many years of hardware and software development that it is highly unlikely any entirely new designs will make up for the software head start that the 68000 and its 32 bit derivative, the 68020, presently enjoy.

The use of this P-Code interpreter would later be rationalized by purveyors of the p-System who emphasized the alleged portability of programs in P-Code form. (Portability refers to the ease of moving a program written on one type of computer over to another and making it run.) The truth is, there is nothing more portable than source code in a reasonably standardized (HLL). ("Source" is what you type into the computer when you sit down and write a program, "Line 10 FOR X = 1 TO 99: NEXT X " etc.)

Source code written for a micro in a given HLL can usually be transported over to a new compiler or interpreter in ASCII form. An editor can then be used to search and replace any differences in syntax or I/O features. There is no speed penalty paid for this kind of portability — but software authors worry about the ease with which source codes can be plagiarized. Authors would rather sell programs in P-Code form which is much harder to decipher and plagiarize.

When the UCSD p-System was written, 32K of RAM was typical for a minicomputer. To avoid problems with such limited memory, program modules were designed to be loaded into the small RAM address space from a fast (expensive) hard disk as needed, a process called "module swapping" or "virtual memory." In this way a large and powerful operating system could be used in a computer with as little as 64K of RAM. Large application programs could also be written in modular fashion by committees of programmers. It is in the nature of the UCSD p-System to allow such modular programs (as well as to link HLLs, such as BASIC, to assembly language). Even individual application programmers prefer to write large business programs in small manageable modules.

On the other hand, operating system program modules called from a hard or floppy disk into a small address space are at a decided speed disadvantage; the procedure is slower than having all the operating system code in RAM at the same time. Therefore, the way to speed up this operating system is to load all those old software modules into an area of RAM called a "RAM disk" or "disk emulator." Fortunately, the 68000 has high-speed machine-language block move instructions to swap modules out of RAM instead of mechanical disk. The UCSD P-System comes alive when it operates entirely in RAM on a fast 68000 and is far superior to any other popular operating system presently available on a micro. It is a real eye opener to have an 80K text file, a powerful program editor and a compiler or assembler in RAM, all at the same time. The age old cycle of compiling your program, finding the errors, reloading the editor, reloading the text file, correcting the error, saving the corrected text file, reloading the compiler and recompiling the program is almost instantaneously achieved by pressing less than half a dozen keys.

The p-System was not always this easily used. The early acceptance of the very large (over 100K) p-System was retarded when Apple Computer tried to use it on an 8 bit 48K Apple II with only 143K on the floppy disk. To get just a directory or catalog on the video screen, the p-System filer program module had to be loaded into RAM from a floppy disk, a very slow and frustrating experience.

Ironically, the p-System worked much better on the Apple III, which had large bank switched RAM space and a hard disk. Not enough Apple IIIs were sold to improve the reputation of the p-System. We believe that the p-System on the Apple III was used by Apple Computer to write (in PASCAL) the operating system for the Lisa. Mike Markula, past president of Apple Computer, mentioned that he thought that the Apple III running Apple's version of the p-System was the best software development system on the market. It probably was at the time, but not for writing operating systems in a HLL. Writing the Lisa's operating system in a high level language was a dumb idea. This is one possible explanation for the original Lisa's slowness.

Large numbers of p-Systems were sold for the Apple II, thereby giving the P-System a reputation as being cumbersome and slow on micros in general. Its implementation in 512K RAM disk on the eight mhz or faster 68000 is helping change this image.

The p-System needs over 100K of memory because it is a very large and comprehensive operating system with a complete complement of programming tools including:

A) A program editor with search-and-replace, block move, forward and backward scrolling and save functions, which is fast and easy to use. For example, in the event of a syntax error during program compilation, the p-System will automatically and quickly, in three or four seconds, reload the source text file from RAM disk and return the cursor to the error on the screen with one or two keystrokes. This is handy for writing compiled programs by trial-and-error. (Don't laugh, it works). The p-System editor is far better than ED on CP/M 68K (which, admittedly, isn't saying

much). It is a real eye opener to have a 100K text file in RAM and be able to jump from the beginning to the end in only three seconds (no mistake — 3 seconds!!!!!). Try that on your Apple II.

At one time we did a lot of 6502 assembly language programming on the Apple II, and we were not satisfied with the speed or the memory capacity. We investigated ways to get all those 6502 assembly language text files out of the Apple II into the Sage II in p-System editor text file format, and finally discovered a way to upload Apple II assembly language text files to the P-System editor. We were then able to cross assemble them after a few changes with the editor. (An unexpected bonus, most welcome.) BASIC and PASCAL text files were also uploaded. The secret to doing this is to use the Apple II serial printer interface and a utility on the p-System called "TEXTIN". The P-System program editor's replace function is easily used to change 6502 assembly language pseudo-ops and Applesoft BASIC commands to conform to p-System language requirements.

B) A general purpose 68000 macro assembler (and cross-macro assembler) which assembles code for almost any microprocessor. Each set of mnemonics is loaded separately.

C) A linker to link assembly language modules, as well as high level language modules including BASIC, to other high level language modules as well as to themselves. You can even link a PASCAL procedure to a BASIC program and use it as a subroutine.

D) A very fast, comprehensive and flexible disk operating system.

E) Compilers for BASIC, FORTRAN, PASCAL and MODULA II, among other HLLs. These compilers compile to P-Code, which can then be partially compiled to 68000 native code using the p-System's Native Code Generator. (Unfortunately at the present time, this Native Code Generator does not compile all P-Code to native code, only selected routines. The P-Code interpreter program is still needed in RAM to handle the remaining routines). A further increase in speed could be achieved by rewriting this Native Code Generator to compile all P-Code to native code. The entire p-System could then be compiled to native 68000 code.

UNIX operating systems are getting a lot of attention in the computer press lately. UNIX has several desirable features as does CP/M-68K. The p-System, however, has much more application software to run on the 68000 than CP/M-68K or any UNIX-like operating system (a reason in itself why the p-System is becoming the de facto standard operating system on the 68000). To my knowledge, there are less than a dozen application programs for CP/M-68K or UNIX. At last count over 250 packages were available for the p-System. Most of this application software for the p-System was first written for Apple IIs with the PASCAL language card and minicomputers running the UCSD p-System. There are several of each, very good wordprocessors, data base and spreadsheet programs. As the use of low cost 68000 computers that use the UCSD p-System grows, the body of copyable software for the p-System should also increase. A 248 page book that lists all the software is available from Softech Microsystems, San Diego, CA., for six dollars.

Although the p-System was written in PASCAL, compiled to P-Code, then interpreted and module-swapped, it runs faster on a 8 MHZ 68000 micro, using a RAM disk, than any UNIX or UNIX-like operating system that we know of. Not only is UNIX a highly modular operating system, but it has the greater disadvantage of having been written on an older, more memory limited, minicomputer than that used to write the p-System. Most — if not all — UNIX-like systems were written in 'C' and compiled to assembly language, then assembled to native machine code (which is in the nature of writing with 'C').

'C' based operating systems should be faster than systems written with a pseudo compiler which uses a P-Code interpreter; therefore, UNIX should be faster than the p-System — but it isn't. One possible reason for this is the fact that UNIX program modules are stored on hard disk in fragmented blocks and swapped in from hard disk to small areas of RAM (virtual memory). Thus the additional unnecessary complexity and number of modules in UNIX make UNIX slower. Maybe it will come alive when 2 megabyte RAM disks are common. UNIX is also a multi-user operating system when what is really needed is a one CPU per user, net-worked operating system. (Single CPU, multi-user systems, went out with high priced, discrete chip central processing units such as found in mainframes and minicomputers. Microcomputers (less disk drives) now cost little more than dumb terminals used with the old multi-user CPUs.) These are some of the reasons that the UNIX running, multi-user, Fortune 500 and the Radio Shack Model 16, 68000 micros are slow, compared to a single-user 68000 running the p-System.

The UCSD p-System in RAM disk on a 68000 is usefully fast; UNIX, at this time, is not. Yet, the p-System's performance could be improved even over its present capabilities if it were rewritten to take advantage of the 68000's large (16 megabyte) lincar address space. Rewriting the p-System would preclude having to swap modules from RAM disk and thereby make it more efficient.

Nevertheless, the p-System is becoming the de facto standard, 68000 personal computer, operating systcm, something the 68000 community needs. There are already a surprising number of low-cost 68000 microcomputers running the UCSD P-System. The following microcomputers deserve serious consideration, described in order of usefulness-per-dollar, according to our own personal preferences. While our tastes may not be the same as everyone's, it would be nice to think that at least we represent a reliable cross-section of knowledgeable micro users:

Sage II

($3200) Sage Computer

Upon reading an ad in "Byte" for the Sage, we contacted Sage Computer for dealer information. We were pleasantly surprised when a knowledgeable salesman appeared and demonstrated the machine, which initially impressed us because it came with p-System, wordprocessing, spreadsheet, PASCAL and a 68000 macro assembler, along with an assortment of other software. When we saw the extensive

documentation, the schematic, the memory map, the powerful system monitor in 16K byte EPROM, and the monitor source listing — in other words, a completely open system — we were sold.

The experience was like that of a few years before, when we were first introduced to the Apple II, except that with the Sage we were given an extensive assortment of software and a built in printer interface just to start up our acquaintance. In short, we bought a Sage and have been pleased with this supermicro to this day; it has proven its reliability and speed.

Floppy disk access and load times (20K per sec.) execute on the Sage about ten times faster than on the Apple II disk operating system (DOS) and BASIC programs run four times faster than on IBM's Personal Computer. It is as fast to program in high level compiler languages as using interpreters on 8-bit machines. Our assembly language programming productivity doubled with the Sage. Word processing spelling checkers, too slow previously, are viably fast.

An unexpanded, 256K Sage II costs approximately $2600 (discounted). You can plug in your own 64K bit dynamic RAM chips for 512K bytes and your own second Mitsubishi floppy disk drive; sockets, cables and connectors are provided with the unexpanded machine. Industry standard, one hundred and fifty nanosecond, 64K bit RAM chips cost about six dollars at the present, and 36 chips make up 256K of parity RAM memory. A Mitsubishi floppy is presently about $350.

The Sage II boasts 24 bit address, 16 bit data bus, expansion connectors. It comes with a built-in Centronics parallel printer port, an IEEE-488 port and two RS-232 serial ports, one of which is used with the terminal, the other already set up for a modem. The standard drive uses five-inch double-density double-sided floppies with 640K on each disk. Very expensive options include hard disk up to forty megabytes and a six-user system with 1 megabyte RAM. Unfortunately the clock is not battery backed up.

Several other operating systems will run on the Sage, including CP/M-68K, Mirage, PDOS, BOS/5 and Idris (a Unix-like operating system). Languages that run under the standard and optional operating systems are several versions of Fortrans, BASIC, ADA, Forth, Cobol, Microcobol, APL, Modula II and several 'C's. The Sage requires a separate RS232 terminal.

Pinnacle

($3895). Pinnacle Systems

The Pinnacle comes with just the p-System run-time package. The rest of the p-System is optional. The Pinnacle is equipped with two double sided, quad density, 800K half height floppy drives. 512K byte parity RAM is standard. RAM expansion to one megabyte costs $1295. Pinnacle is extremely cost effective particularly in the 10 MEG hard disk version at only $5995.

[Editor's Note: The authors included two paragraphs questioning Pinnacle's claim to run at 12 Mhz. We contacted Pinnacle and they stated that "... the Pinnacle runs at 12 Mhz and has done so since June of 1983. This is possible by using prequalified selected parts."]

The Pinnacle has no less than seven serial ports, one parallel printer port, an Omninet networking bus and a Sasi hard disk interface in the basic machine, as well as a battery-backed-up clock and calendar.

The video display and keyboard aren't built-in on the Pinnacle, unlike the Apple II; a separate RS232 serial terminal is required. However, not having a built-in display and keyboard can be advantageous, because the buyer only pays for what he needs. Separate 19.2K baud serial terminals are also faster than most built-in HiRes bit-mapped displays (another reason the 68000 bit-mapped Apple Lisa is so slow?). This is due to the dedicated CPU in all terminals (multi-processing if you will) that has nothing else to do but update the screen while bit mapped displays are usually updated by the main CPU. An RGB color terminal costs about the same as an RGB board and color monitor for the Apple II or the IBM-PC, and this way the buyer can choose according to preference. Something not to overlook is that most microcomputers can be used as terminals.

The question being asked these days is, "Will the company be here next year?" Pinnacle Systems is part of a larger company that specializes in manufacturing complete computer boards. The excellent design of the four layer Pinnacle processor board is testimony to their expertize in this area.

Dimension 68000

(Under $4000) Micro Craft Corporation

This machine reputedly took 16 months to engineer. Its 68000 chip runs at 7.2 Mhz and optionally runs the p-System. Idris, Mirage, Unix 5, S1 and Concurrent DOS 4.0 will be available soon. CP/M-68K is standard. The amazing attribute of the Dimension is that it will run almost any other company's or microprocessor's software. Micro Craft has engineered plug-in CPU cards that not only run the software of other popular machines, but automatically reconfigure the disk drive format and video display to imitate the hardware configuration of the simulated machine. In other words, when imitating an Apple II, the disk drives have 143K and the display has 40 characters across, upper case only.

The display can handle up to 100 characters across and up to 48 lines. There is a color graphics mode, 160 X 480 pixels in 16 colors and a black and white graphics mode of 640 X 480 pixels. 512K of RAM can reside on the main board. The Dimension comes with 256K of RAM. Chips can be plugged in to increase memory up to 512K. A six-slot internal expansion but is provided. 3.5 megabytes (!) of RAM can be plugged into these expansion connectors, using 64K bit dynamic RAM chips on 512K byte plug-in cards costing $1295 each. When 256K bit dynamic $37 RAM chips become lower in cost (next month), 12.5 megabytes in the box are possible, although the power supply might have to be changed to install 12.5 megabytes of RAM. An optional expansion interface will allow another 4 megabyte of directly addressable RAM for a total of 16MB.

To put 16 megabytes of RAM in perspective, the largest and most expensive supermini that the Digital Equipment Corporation manufactures is the VAX 11/782, which costs about twice as much as a VAX 11/780 and the maximum amount of memory it can address is 8 megabytes (Mini-Micro Systems magazine.)

Extra-cost half-height five inch double-sided 800K byte floppies are available. The standard machine comes with two 5-1/4 inch, 400K byte floppies; serial, parallel and game controller interfaces. A 50 megabyte hard disk and controller costs $4995.

Hewlett Packard 9816A

($3995) Hewlett-Packard

The HP 9816A is a 68000 personal computer with a too small, nine inch CRT having a 400 X 300 pixels graphic capability. The indispensable disk drives are a lot extra. The HP 9816A uses the 270K Sony three-and-a-half inch floppy disk drives at only $900 each, but the hard-shell media for this drive are presently hard to find and expensive. The use of these drives and media on the Apple Macintosh will probably change this. Dual 540K, five inch drives cost $2230. The same Tandon drives bought "off the shelf" from an electronics distributor cost $250 each.

The HP 9816A runs the UCSD p-System version 2.1 (everybody else uses version 4.13) at an extra $1515. There are 128K RAM, a well thought out RS232 and HP-IB (IEEE 488) interfaces built in. Additional RAM cost about $1100.00 for 256K, while on the Sage or the Dimension 256K RAM would cost $300 to $650. There are various other well engineered, expensive options on the HP 9816A such as plotters and graphics tablets. Apparently, HP means High Price.

Corvus Concept

($3995) Corvus Systems

The UCSD p-System runs on the Corvus Concept as a $695 option. The CPU is an 8 mhz 68000. It has a large (15 inch) crt display, 132 columns by 66 lines and 720 X 560 HiRes graphics. The display is black and white only. The bit-mapped design makes the display update slower than a 19.2K baud terminal. A large speed improvement could be made by Corvus if they put a dedicated micro in the display with the sole purpose of updating the display.

Believe it or not, the Corvus Concept has a 50 pin Apple II expansion bus! There are two built-in RS232 serial interfaces and an Omninet interface. The Corvus also has a battery-backed-up clock and calendar and 512K of RAM in the box. From a marketing viewpoint, a weakness of the Concept is that it doesn't come with any floppy disk drives. Single 750K five inch floppies are a lot extra ($750). A further handicap is a rather slow access time (1 megabit per second) Omninet, networked hard disk. A hard disk with an eight bit parallel interface is also available that should be somewhat faster.

However, the Corvus Concept is worth considering if the system must support more than about four users, because it can utilize a network. Multi-using eventually boggs down any single CPU, even the 68000. By comparison, Digital Equipment Corporation offers the VAX 11/780 with 96 users. Can you imagine 96 users, all trying to compile different programs at the same time, on one CPU no faster than a fifty dollar, 12 mhz 68000 microprocessor? (Maybe Digital doesn't think all 96 users are going to be on-line at the same time.) Multi-user, single CPU time-sharing systems are simply obsolete. Networked microprocessor CPUs are the only answer.

Saybrook

($995) Analytical Engines

This is an add-on 68000 computer for the Apple II. It has an 8 mhz or optional 12.5 mhz CPU and 128K of RAM on-board, expandable to 512K with 256K bit chips. The Apple II quickly communicates with the 68000 computer by means of DMA (direct memory access). The weakness of this approach is the Apple's slow disk operating system and the limited storage capacity 143K Apple disk drives. On the other hand, program execution and video screen updates are very fast, particularly for HiRes animated graphics. (The Apple II has a rather unique and very fast screen update scheme. In effect it is a dual processor. The first is the main 6502 CPU. The second, very simple processor, made from discrete TTL chips, has nothing else to do but refresh the screen and the dynamic memory.) Analytic Engines claims 10 to 30 times the speed of the Apple II with the 68000 running programs out of RAM dedicated to the 68000.

The computer comes with the p-System and a choice of one of the following languages: UCSD PASCAL, FORTRAN, BASIC compilers or an Applesoft-compatible BASIC interpreter. The Saybrook has a 24-hour clock (but it's not certain that this clock is backed up by a battery).

These are the presently available low-cost 68000 microcomputers running the UCSD p-Systems. In the near future other 68000 computer manufacturers will probably hitch a ride on the p-System bandwagon, the mouse and Apple's Macintosh to the contrary. The Apple Lisa II would be a fine candidate as would the Radio Shack Model 16 and the Fortune.

You have to try the p-System on a fast 68000 machine with RAM disk to understand the difference between the various other operating systems and the very real advantages offered by the p-System and a fast and powerful 68000. Any of the above microcomputers (where available) are worth a trip to the computer store.

Manufacturer's References

Analytical Engines
3415 Greystone, #305
Austin, TX 78731
512/346-8430

Corvus Systems
2100 Corvus Drive
San Jose, CA 95124
408/559-7000

Hewlett-Packard
19447 Pruneridge Avenue
Cupertino, CA 95014

Micro Craft Corporation
4747 Irving Blvd, Suite 241
Dallas, TX 75247
214/630-2562

Pinnacle Systems
10410 Markison Road
Dallas, TX 75238
214/340-4941

Sage Computer
4905 Energy Way
Reno, NV 89502
702/322-6868

Atari/Epson Custom Characters

Transfer almost unlimited customized alphabets to paper - accurately.

by Mike Bassman

Requirements: Atari 400/600/800/1200, 850 Interface, Epson MX-80 with Graftrax +, or MX-80 FT or MX-100, or FX-80

When it comes to dealing with text, the Atari computers have a marvelous flexibility. Naturally, they can display the usual upper and lower case, numbers and punctuation. Besides that, they also have inverses of all the standard characters, plus lines, card suits and a host of other graphics characters. If you're not satisfied with this selection, you can make your own custom characters which has led to programs using gothic, script and other interesting fonts. You can generate your own fonts with one of the many character-editing programs that have been published. All these can easily be displayed on the screen, but transferring them to paper is not normally possible.

Why? There are a number of reasons. The main one is that printers are not designed for any one particular computer. Only alphanumerics and punctuation symbols are in the standard ASCII table. The maximum possible number of characters is 256 (each character stored uses one byte; a byte can be in the range of 0-255, hence, 256 possibilities). These standard characters fill up less than half of the available room, so Atari decided to pack the rest with inverse and graphics characters. Radio Shack, instead, throws in a combinations of block characters. Commodore has inverse characters and different graphics characters. The point here is that apart from normal characters, no two computers have the same set of 'extra' characters. As such, a printer manufacturer catering only to one computer would have a limited audience. Epson has a viable solution; they have their own characters, an italic set and a few graphics characters. The important fact is that the Epson printers have full graphics abilities. We can take advantage of this to generate Atari's own special and custom characters. All you need is the appropriate software. I've included listings of two somewhat similar programs; one useful and one frivolous.

Any-Text File Lister

The program shown in Listing 1 lists files to an Epson printer. At first this may not sound amazingly useful. I mean, from Basic, this is merely a matter of issuing a LIST "P:" command. From the DOS utilities menu you can copy a text file to P:. but what makes this program useful is that its listing is accurate: it includes all the graphics and inverse characters. If you've ever tried listing a program with graphics or inverse, you'll know that inverse shows up as italics and graphics characters show up as meaningless garbage, or some odd control character will throw the printer into a stupor that will mess up the rest of the listing. This program lists out a program in its exact form, graphics and all. If examined closely, you can see that this program was used to list itself.

Using this program is simple. If there is a Basic program you wish to list, load it in from disk (or cassette) and then re-save it out under a different name using a LIST "D:name" command, rather than the SAVE command. The purpose behind this is to have the program as text, rather than encoded in Basic keywords. If it's a text file you want to print, you clearly don't have to do this. Next run this program and enter the name of the file your text is in. That's all it takes.

Custom Font Message Printer

The program in Listing 2 will print anything you want in a custom character set. If you've ever had a desire to see a message in script or computer-type letters or whatever, this will do the trick. All you need to have are the custom fonts stored on disk (or cassette) in a nine sector file as generated by Instedit (APX) or just about all of the other character generators. There are a few examples of its handiwork shown in the accompanying chart. From top to bottom, the alphabet is shown in computer-style, gothic, fancy fonts and a few others. Making use of this program (custom font message printer) is even easier than the file lister above. Just run it and, when prompted, enter the name of your character set and then the message you want printed. If your character set is stored on cassette, type C: for the character set name, when asked.

Custom Font Variant

Listing 4 shows a program that looks a good deal like the custom font message printer. In fact, it is a cross between that program and the program lister. It does the same thing as the custom font program except that it prints out a whole file in the new font rather than a one line message. This would be useful to take a file generated with a word processing program and, after putting it through this program, end up with a professional looking document printed

in a pleasant typeface of your choice. There are some commercially available programs which do just this. Using it consists simply of entering the name of the font and the name of the file to be printed.

Entering The Programs

Typing in these programs can be a bit of a problem because of the machine language subroutine embedded in them. The Basic part is easy to do. The straightforward approach is to type those graphics characters just as you see them. The graphics keyboard included in the Atari Basic Reference manual is a guide to finding all the right keys (the back cover has a relatively easy to use diagram). If you do decide to do it this way, make sure that you save the program before running it. Any typo could bomb the computer; let this be a word to the wise.

Another method is to assemble the source code using the Assembler/ Editor cartridge or one of the many other assemblers available. The source code for the machine language portion is shown in Listing 3. The programs all use the same machine language subroutine, so the most difficult part only has to be done once, even if you want all the programs. The amount of code needed to be typed in this way is longer than typing the graphics characters, but fortunately the code is made up of normal alphanumeric characters. You then assemble the code to a disk file and modify either of the programs to load in this subroutine from that disk file. If you choose to do it this way (not highly recommended), the changes to listings 1 & 2 are as follows:

1 - Delete lines 11,12,13.
2 - Add lines 90-130 as shown.

```
90 OPEn #3,4,0,"D:ASSEM.OBJ"
100 FOR K = 1 TO 6:GET #3,X:NEXT K
110 TRAP 130:K = 0
120 K = K + 1:GET #3,X:
ML$(K,K) = CHR$(X):GOTO 120
130 CLOSE #3
```

The modification here can be used with both programs and with the variant by changing the line numbers. The filename in line 90 is your assembled version of the source code. These programs are very similar, so if you want to have all of them, I would recommend typing in one of them, saving it, and then modify it until you have the other program. Conversely, if you're only bothering with one program, keep in mind it doesn't take much effort to obtain the other ones.

Theory of Operation

You already have all you need to know to get these programs working. But if you want to know how they work, and maybe do clever things of your own with your Epson, read on.

You may ask how we get a large number of new character sets out of a printer normally limited to regular and Italic characters, in various sizes. Well, these programs don't exactly print out new characters, they draw them. All reasonably new Epson printers (or old ones retrofitted with Graftrax +) have the ability to do graphics. There's no reason to limit use of the graphics to charts or drawings; you can also improve on your regular text performance. The basic task to be accomplished is to get a character from the file, find out how the character is drawn, send this information over to the Epson and repeat this procedure until you reach the end of the file (or message).

Where Character Shapes Are

The character set in use can be found at the address specified by **PEEK(756) *256**. Location 756 (2F4 hex) is the Character Base register, holding the high byte of the address. The low byte is assumed to be zero. The standard character set is at $E000. Though the topic has been more comprehensively covered in other articles, let me quickly refresh your memory on how they (the characters) are stored. Each character can be 8 bits wide and 8 bits high, total of 8 bytes (64 bits). Usually there is a little room on top, bottom and the sides so that characters won't be squeezed too tightly together when shown next to each other. Each row of a character is one byte and there are eight rows going from the top to bottom of any one character. So the capital letter "E" is represented in memory like so:

```
00000000    byte = $00
01111110    byte = $7E
01100000    byte = $60
01111100    byte = $7C
01100000    byte = $60
01100000    byte = $60
01111110    byte = $7E
00000000    byte = $00
```

Getting a Character to the Printer

This is how the computer reads a normal or redefined character. Now we've got to send this information out to the printer. Things would be simple if the printer could be fed the character a byte (row) at a time, just like the computer understands them. But this isn't the case. While the computer reads a character a row at a time, from top to bottom, the printer head is a vertical column, so it does each character a column at a time, from left to right. This makes life difficult. What we're going to have to do is take each byte that forms a row of the character and take off the leftmost bit. We're going to take these bits off all eight rows, line them up in a column and then send the column off to the printer. Then we do this for the eight columns that make a character, from left to right. Visually, this means that instead of taking slices of bits off the top of say, that 'E' we saw earlier, we take slices vertically off the sides.

Theory into Code

This cut and paste type of operation with bits can be turned into a basic program. To output one complete character, we need two loops, one going from left to right sending out columns of data and an inner loop that puts together these columns. There's a chart showing each pin of the print head and what is needed to turn it on:

```
128 - o
 64 - o
 32 - o
 16 - o
  8 - o
  4 - o
  2 - o
  1 - o
```

For each of the pins you want to turn on, add that number. For example sending a 34 would turn on the third pin from the top and the second from the bottom. It's no surprise that each of these values is 2 to the power of the pin number (pins numbers range from 0 to 7, bottom to top), and we'll use this fact. To find out if we want to turn on a pin, we look at a row of the character, AND it with the column number we're up to and, if we get a positive value, we know to turn it on. Column numbers, not coincidentally, are represented just like the pin numbers but from right to left, instead of bottom

Listing 2

```
                ; CUSTOM CHARACTER DUMP
                ; MIKE BASSMAN
                ;
Ø342            ICCOM    EQU $342
Ø344            ICBAL    EQU $344
Ø345            ICBAH    EQU $345
Ø348            ICBLL    EQU $348
Ø349            ICBLH    EQU $349
Ø68Ø            B        EQU $68Ø
Ø681            G        EQU $681
Ø682            SUM      EQU $682
ØØ2C            VL       EQU $2C
Ø685            J        EQU $685
Ø687            AD       EQU $687
Ø689            VALUE    EQU $689
Ø68A            MASK     EQU $68A
Ø68B            RESULT   EQU $68B
Ø68C            INVRS    EQU $68C
                ;
5ØØØ                     ORG $5ØØØ
                ;
                ; SAVE NUMBER OF ARGUMENTS
                ; CHAR MEM HI
                ; CHAR MEM LO
                ; SET INVERSE VIDEO FLAG
                ;
5ØØØ 68         START    PLA
5ØØ1 68                  PLA
5ØØ2 8D 88 Ø6            STA AD+1
5ØØ5 68                  PLA
5ØØ6 8D 87 Ø6            STA AD
5ØØ9 68                  PLA
5ØØA 68                  PLA
5ØØB 8D 85 Ø6            STA J
5ØØE A9 ØØ               LDA #$ØØ
5Ø1Ø 8D 8C Ø6            STA INVRS
5Ø13 AD 85 Ø6            LDA J
5Ø16 C9 8Ø               CMP #$8Ø
5Ø18 9Ø ØA               BCC INIZ
5Ø1A 29 7F               AND #$7F
5Ø1C 8D 85 Ø6            STA J
5Ø1F A9 FF               LDA #$FF
5Ø21 8D 8C Ø6            STA INVRS
                ;
                ; SETUP ROW/COLUMN COUNTERS
                ;
5Ø24 A9 Ø7      INIZ     LDA #7
5Ø26 8D 8Ø Ø6            STA B
5Ø29 A9 ØØ      FIRSTG   LDA #Ø
5Ø2B 8D 81 Ø6            STA G
5Ø2E 8D 82 Ø6            STA SUM
5Ø31 A9 ØØ      SETVL    LDA #Ø
5Ø33 85 2C               STA VL
5Ø35 85 2D               STA VL+1
5Ø37 AD 85 Ø6            LDA J
5Ø3A A2 Ø3               LDX #3
                ;
                ; MULTIPLY BY 8
                ;
5Ø3C 18         MULT8    CLC
5Ø3D Ø6 2D               ASL AL+1
5Ø3F ØA                  ASL A
5Ø4Ø 9Ø Ø2               BCC LOWDO
5Ø42 E6 2D               INC VL+1
5Ø44 CA         LOWDO    DEX
5Ø45 DØ F5               BNE MULT8
5Ø47 85 2C               STA VL
5Ø49 18                  CLC
5Ø4A 6D 87 Ø6            ADC AD
5Ø4D 85 2C               STA VL
5Ø4F 9Ø Ø2               BCC ADDHI
5Ø51 E6 2D               INC VL+1
5Ø53 18         ADDHI    CLC
5Ø54 A5 2D               LDA VL+1
5Ø56 6D 88 Ø6            ADC AD+1
5Ø59 85 2D               STA VL+1
5Ø5B 18                  CLC
5Ø5C A5 2C               LDA VL
5Ø5E 6D 81 Ø6            ADC G
5Ø61 85 2C               STA VL
5Ø63 9Ø ØA               BCC ANDIT
5Ø65 E6 2D               INC VL+1
5Ø67 BØ Ø6               BCS ANDIT
5Ø69 18         ELONG    CLC
5Ø6A 9Ø C5               BCC SETVL
5Ø6C 18         LONG     CLC
5Ø6D 9Ø BA               BCC FIRSTG
5Ø6F AØ ØØ      ANDIT    LDY #Ø
5Ø71 B1 2C               LDA (VL),Y
5Ø73 4D 8C Ø6            EOR INVRS
5Ø76 8D 89 Ø6            STA VALUE
5Ø79 AE 8Ø Ø6            LDX B
5Ø7C EØ ØØ               CPX #Ø
5Ø7E DØ Ø5               BNE SMALL
5Ø8Ø A9 Ø1               LDA #1
5Ø82 18                  CLC
5Ø83 9Ø Ø8               BCC AMAIN
5Ø85 A9 Ø2      SMALL    LDA #2
5Ø87 CA         DO       DEX
5Ø88 FØ Ø3               BEQ AMAIN
5Ø8A ØA                  ASL A
5Ø8B DØ FA               BNE DO
5Ø8D 8D 8A Ø6   AMAIN    STA MASK
5Ø9Ø 2D 89 Ø6            AND VALUE
5Ø93 8D 8B Ø6            STA RESULT
5Ø96 C9 ØØ               CMP #Ø
5Ø98 FØ 1F               BEQ NEXTG
5Ø9A 38                  SEC
5Ø9B A9 Ø7               LDA #7
5Ø9D ED 81 Ø6            SBC G
5ØAØ C9 ØØ               CMP #Ø
5ØA2 DØ Ø5               BNE POWER
5ØA4 A9 Ø1               LDA #1
5ØA6 18                  CLC
5ØA7 9Ø Ø9               BCC DOSUM
5ØA9 AA         POWER    TAX
5ØAA A9 Ø2               LDA #2
5ØAC CA         DO2      DEX
5ØAD FØ Ø3               BEQ DOSUM
5ØAF ØA                  ASL A
5ØBØ DØ FA               BNE DO2
5ØB2 18         DOSUM    CLC
5ØB3 6D 82 Ø6            ADC SUM
5ØB6 8D 82 Ø6            STA SUM
5ØB9 EE 81 Ø6   NEXTG    INC G
5ØBC A9 Ø8               LDA #8
5ØBE CD 81 Ø6            CMP G
5ØC1 DØ A6               BNE ELONG
5ØC3 A2 4Ø               LDX #$4Ø
5ØC5 A9 ØB               LDA #$B
5ØC7 9D 42 Ø3            STA ICCOM,X
5ØCA A9 82               LDA #SUM
5ØCC 9D 44 Ø3            STA ICBAL,X
5ØCF A9 Ø6               LDA /SUM
5ØD1 9D 45 Ø3            STA ICBAH,X
5ØD4 A9 Ø1               LDA #1
5ØD6 9D 48 Ø3            STA ICBLL,X
5ØD9 A9 ØØ               LDA #Ø
5ØDB 9D 49 Ø3            STA ICBLH,X
5ØDE 2Ø 56 E4            JSR $E456
5ØE1 18         NEXTB    CLC
5ØE2 CE 8Ø Ø6            DEC B
5ØE5 1Ø 85               BPL LONG
5ØE7 6Ø                  RTS
5ØE8                     END
```

to top. This strategy is represented in the following piece of Basic-like code.

J = ASC(CHARACTER)
A = ADR(STARTOFCHARACTER-DATA)
FOR B = 7 TO 0 STEP -1:REM the outer column loop.
SUM = 0:REM clear the print head counter.
FOR G = 0 TO 7:REM inner loop totals up a column.
Y = PEEK(A + G + J*8):REM get the row value.
X = INT(2 B + .5):REM the column number.
Z = X AND Y:REM you can't do a boolean AND in Basic, but you get the idea.
IF Z THEN SUM = SUM + INT(2 (7-G) + .5):REM add pin value to running total if we should.
NEXT G:REM do it for the entire column.
PUT #4,SUM:REM output the column to printer.
NEXT B:REM now do this for all 8 columns.

This is ridiculously slow when done in Basic, so the machine language subroutine just uses this algorithm, but runs infinitely faster. There is only one other thing you need to know to control the Epson. Before you start sending all this pin information, you have to tell it to go into high resolution mode and then say how many columns of graphics you want. Turning on graphics is done by sending an ESCAPE, then a "k". You tell it how many columns of graphics by sending out two more values, the first being the low byte of the # of columns, the second being the high byte.

That's all there is to making your Epson print anything you want. The programs listed here are only a few of the possible applications. Using some of the information shown here, you can invent new and interesting uses for your printer.

```
13 ML$(198)="▮ ▀▚B┛▮▀ ▚D┛▮/▚E┛▮ ▚H┛▮♥▚I┛  V▚┼▚▚/♣▀♦"
17 I=0:DIM X$(11),M$(255)
20 X$="h▀♥/-├/▀ ▚/♦"
30 A=PEEK(756)*256
40 DIM N$(15),NM$(15)
50 GRAPHICS 18
60 POSITION 5,1:? #6;"ANY TEXT"
70 POSITION 2,5:? #6;"program lister"
80 POSITION 5,9:? #6;"for epson"
```

Printer Sample

Listing 1

```
10 DIM ML$(300)
11 ML$(1)="hh▀/h▀\/hh▀▚/▮♥▀/▀▚/▚♥▚) ▶▀▚/▮▶▀/▮\▀▚/▮♥▀▚/▀/▮♥▚,▚-▀▚/▚┼/-▚▚ ▚-▚▚▚,┼▀\/▚,▚ ▚-┼▚-▀/▚-"
12 ML$(92)="┼▚,▀▚/▚,▚▚-▚/┼▚┼▚▀♥▚,▀▚/▀/▚/▚♥▚▮┼▚▚ ▚▚◣▚▚/-▀/▀/▚♥▚+8▮\▚/▚♥▚▮┼▚▚▚ ▚▚◣▚┼▀/▀/▚/▮▚/▚e"
13 ML$(198)="▮ ▀▚B┛▮▀ ▚D┛▮/▚E┛▮ ▚H┛▮♥▚I┛  V▚┼▚▚/♣▀♦"
17 I=0:DIM X$(11),M$(255)
20 X$="h▀♥/-├/▀ ▚/♦"
30 A=PEEK(756)*256
40 DIM N$(15),NM$(15)
50 GRAPHICS 18
60 POSITION 5,1:? #6;"ANY TEXT"
70 POSITION 2,5:? #6;"program lister"
80 POSITION 5,9:? #6;"for epson"
90 FOR K=1 TO 2000:NEXT K
190 GRAPHICS 0:POSITION 2,6
200 ? "Please enter the name of the disk      file which contains the program to be listed.  (ex: PROG.BAS)"
210 ? :? "Name of program:"
220 TRAP 350:NM$="D:":INPUT N$:NM$(LEN(NM$)+1)=N$
230 OPEN #3,4,0,NM$
240 INPUT #3,M$
270 TRAP 400
285 Q=LEN(M$):IF LEN(M$)>60 THEN Q=60
290 I=0:J=Q*8:IF J>255 THEN I=1:J=J-256
295 OPEN #4,8,0,"P"
300 PUT #4,27:PUT #4,ASC("K"):PUT #4,J:PUT #4,I
310 FOR K=1 TO Q
320 J=ASC(M$(K,K)):IF J<96 AND J>31 THEN J=J-32:GOTO 330
322 IF J<32 THEN J=J+64:GOTO 330
324 IF J<224 AND J>159 THEN J=J-32:GOTO 330
326 IF J<160 AND J>127 THEN J=J+64
330 DUMMY=USR(ADR(ML$),A,J)
340 NEXT K:PUT #4,27:PUT #4,64:CLOSE #4:LPRINT
342 IF Q<LEN(M$) THEN M$=M$(61):GOTO 285
345 TRAP 350:GOTO 240
350 CLOSE #3:CLOSE #4:GRAPHICS 18
360 POSITION 6,4:? #6;"all done"
370 FOR K=255 TO 0 STEP -1:SOUND 0,K,12,5:SOUND 1,255-K,12,3:NEXT K:SOUND 1,0,0,0
380 GRAPHICS 0:END
390 ? :? "KERROR IN LOADING CHARACTER SET":? :CLOSE #3:GOTO 200
400 ? "KERROR: PRINTER OR INTERFACE NOT READY"
410 END
420 ? "KERROR IN OPERATION - PLEASE TRY AGAIN"
430 END
```

How about a $2000 TYPEWRITER !

Printer Sample

SORT OF A BIZZARE 1960'S HIPPY TYPE STYLE

TRANSLYVANIA GOTHIC TIMES - Midnight Edition

THE COMPLETE OUTLINE OF ALL COMPUTER KNOWLEDGE

Listing 3

```
10 DIM ML$(300)
11 ML$(1)="hh[illegible]"
12 ML$(92)="[illegible]"
13 ML$(198)="[illegible]"
17 I=0:DIM X$(11),M$(60)
20 X$="h[illegible]"
30 DIM B$(1024):A=ADR(B$)
40 DIM N$(15),NM$(15)
50 GRAPHICS 18
60 POSITION 4,1:? #6;"CUSTOM FONT"
70 POSITION 2,5:? #6;"message printer"
80 POSITION 5,9:? #6;"for epson"
100 FOR K=1 TO 2000:NEXT K
140 GRAPHICS 0:POSITION 2,6
150 ? "Remember: maximum of 60 characters"
160 ? :? "Please enter message you wish to print"
170 ? "Message:":TRAP 140:INPUT M$
180 IF LEN(M$)=0 THEN 140
190 GRAPHICS 0:POSITION 2,6
200 ? "Please enter the name of the disk      file which cont
ains the character fontto be used (ex: GOTH.SET)"
210 ? :? "Name of character set:"
220 TRAP 390:NM$="D:":INPUT N$:NM$(LEN(NM$)+1)=N$
230 OPEN #3,4,0,NM$
240 FOR K=0 TO 1023
250 GET #3,B
260 POKE A+K,B
270 NEXT K:CLOSE #3:TRAP 400
280 OPEN #4,8,0,"P":TRAP 420
290 J=LEN(M$)*8:IF J>255 THEN I=1:J=J-256
300 PUT #4,27:PUT #4,ASC("K"):PUT #4,J:PUT #4,I
310 FOR K=1 TO LEN(M$)
320 J=ASC(M$(K,K)):IF J<96 AND J>31 THEN J=J-32
330 DUMMY=USR(ADR(ML$),A,J)
340 NEXT K:PUT #4,27:PUT #4,64:CLOSE #4:LPRINT
350 GRAPHICS 18
360 POSITION 6,4:? #6;"all done"
370 FOR K=255 TO 0 STEP -1:SOUND 0,K,12,5:SOUND 1,255-K,12,3
:NEXT K:SOUND 1,0,0,0
380 GRAPHICS 0:END
390 ? :? "ERROR IN LOADING CHARACTER SET":? :CLOSE #3:GOTO
200
400 ? "ERROR: PRINTER OR INTERFACE NOT READY"
410 END
420 ? "ERROR IN OPERATION - PLEASE TRY AGAIN"
430 END
```

How would you like to have an entire issue of MICRO printed in a GOTHIC font? While it might be interesting at first, I am sure that the novelty would wear off very quickly.

Printer Sample

Listing 4

```
10 DIM ML$(300)
11 ML$(1)="hh▄◤/h▄◥hh▄▜/▐♥▄▟/▄▜/▐◤◣)▸▄▜/▐▐▄▟/▐◥▄◤/▐♥▄▟/▄▟/
▐♥▜,▜─▄▜/▟┘┴/─◣◲ ▟─▟▟▜,┴▄◥▜,◲ ▟─┴◲─◤/▜─"
12 ML$(92)="┴◲,▄▟/▜,◲◣▟─▢/┴◲◲┴◲:▄♥▐,▄▟/▄◤/▄◤/▢♥▟┐▐├─◲◢▐ ▟▟┘◣
▟Z▄◥/─◤/▄▟/▐♥▟→8▐◥▐▟/▐♥▟┐▐├─◲ ▄▐ ▟▟┘◣▟Z┴▄▟/▄▟/▐▟/▐◢▐▟/▟&▟e"
13 ML$(198)="▐ ▟B┘▐▄▟D┘▐/▟E┘▐├▟H┘▐♥▟I┘ V▟┴▟◤/◲▜♦"
17 I=0:DIM X$(11),M$(255)
20 X$="h▄♥/─├/▄ ▙/♦"
30 DIM B$(1024):A=ADR(B$)
40 DIM N$(15),NM$(15),MM$(15)
50 GRAPHICS 18
60 POSITION 4,1:? #6;"CUSTOM FONT"
70 POSITION 2,5:? #6;"Message printer"
80 POSITION 5,9:? #6;"for epson"
100 FOR K=1 TO 2000:NEXT K
110 ? "Please enter the name of the disk      file which cont
ains the character fontto be used (ex: GOTH.SET)"
120 ? :? "Name of character set:"
130 TRAP 390:NM$="D:":INPUT N$:NM$(LEN(NM$)+1)=N$
140 OPEN #3,4,0,NM$
150 FOR K=0 TO 1023
160 GET #3,B
170 POKE A+K,B
180 NEXT K:CLOSE #3:TRAP 400
200 GRAPHICS 0:POSITION 2,6
210 ? "Please enter then name of the disk      file to be prin
ted. (ex: FILE.BAS)"
220 ? :? "Name of program:"
230 TRAP 350:MM$="D:":INPUT N$:MM$(LEN(MM$)+1)=N$
240 OPEN #3,4,0,MM$
250 INPUT #3,M$
260 TRAP 400
270 Q=LEN(M$):IF LEN(M$)>60 THEN Q=60
280 OPEN #4,8,0,"P":TRAP 420
290 I=0:J=Q*8:IF J>255 THEN I=1:J=J-256
300 PUT #4,27:PUT #4,ASC("K"):PUT #4,J:PUT #4,I
310 FOR K=1 TO Q
320 J=ASC(M$(K,K)):IF J<96 AND J>31 THEN J=J-32
324 IF J<224 AND J>159 THEN J=J-32:GOTO 330
326 IF J<160 AND J>127 THEN J=J+64
330 DUMMY=USR(ADR(ML$),A,J)
340 NEXT K:PUT #4,27:PUT #4,64:CLOSE #4:LPRINT
342 IF Q<LEN(M$) THEN M$=M$(61):GOTO 270
345 TRAP 350:GOTO 250
350 CLOSE #3:GRAPHICS 18
360 POSITION 6,4:? #6;"all done"
370 FOR K=255 TO 0 STEP -1:SOUND 0,K,12,5:SOUND 1,255-K,12,3
:NEXT K:SOUND 1,0,0,0
380 GRAPHICS 0:END
390 ? :? "▸ERROR IN LOADING CHARACTER SET":? :CLOSE #3:GOTO
110
400 ? "▸ERROR: PRINTER OR INTERFACE NOT READY"
410 END
420 ? "▸ERROR IN OPERATION - PLEASE TRY AGAIN"
430 END
```

Extended Precision Arithmetic in BASIC

Greater mathematical precision and a way to calculate the lunar-based Jewish Calendar.

by Rolf B. Johannesen

Many common implementations of BASIC in microcomputers today use a binary representation for real numbers which has either 24 or 32 bits for the mantissa and 7 bits for the characteristic. This translates to either 6.1 or 9.5 decimal digits of precision, respectively. Occasionally, greater precision is required: statistical calculations are notable in requiring many extra digits of precision during the intermediate stages of calculations because so many results are derived as the differences between two numbers that are almost equal, so that several of the most significant digits are lost in a single step.

Computer software for processing arithmetic statements never warns that bits have overflowed the mantissa, even though this will inevitably result in loss of precision. However, overflow of bits in the characteristic is always flagged. The program in Listing 1 will test any computer for the length of mantissa in its floating point representation and report it both in terms of bits and equivalent number of decimal digits. The largest integer that can be faithfully represented has a mantissa of all '1' bits, and is equivalent to 2↑M1 if there are M bits in the mantissa. The program will also test the number of bits used for the characteristic. In this case the program will be interrupted at the occurrence of floating point overflow. If your computer does not support the TRAP (or the equivalent ON ERROR GOTO) command in line 210, then line 270 will never be reached. However, the last value of I printed before overflow occurs is the number of bits in the characteristics; the last value of X is a trifle greater than half the largest possible number for that machine. The largest possible number, when there are N bits in the characteristic and M bits in the mantissa, is 2↑(2↑N-1) multiplied by a fraction, very nearly unity, whose numerator contains M '1' bits and whose denominator contains a '1' bit followed by M '0' bits. With a 7-bit characteristic and a 32-bit mantissa, this is very nearly 1.70141183E+38. The hexadecimal representation may vary slightly among BASIC interpreters due to differences in characteristic biasing and in the way the sign bit is expressed. In Microsoft BASIC, the largest possible number has the hex value $FF7FFFFFFF. If your machine lets you alter a number in BASIC's variable table, via monitor or POKEs, you can enter the above value and return to BASIC to print its decimal equivalent.

Extended precision routines in assembly language are perfectly straightforward, rapid and effective; though they tend to get messy for multiplication and especially so for division. Nevertheless, if extensive calculations are required, this method is recommended as being the fastest. It is possible to achieve workable results in BASIC by the procedure given here, in which a large number is broken up and arithmetic operations carried out on the separated parts, with the results combined at the end. If it is necessary for the final result to have greater precision than is available in BASIC, then it will have to be expressed in parts, but this is entirely feasible.

In brief, a large number is expressed as (M*10↑6+T*10↑3+U), where M is the coefficient of the millions place, T the coefficient of the thousands place, and U the units. Obviously, this scheme can be extended to both larger and smaller numbers by choosing the proper powers of ten as multipliers. In

```
10  REM PROGRAM TO MEASURE MAXIMUM PRECISION AND MAGNITUDE OF
20  REM REAL NUMBERS> WHEN ADDING 1 TO OR SUBTRACTING 1 FROM A GIVEN
30  REM NUMBER FAILS TO GIVE A RESULT 1 DIFFERENT FROM THE STARTING
40  REM NUMBER, THE LIMIT OF PRECISION HAS BEEN REACHED. WHEN OVERFLOW
50  REM ERROR OCCURS, THE MAXIMUM MAGNITUDE HAS BEEN REACHED.
100 FOR I=1 TO 100
110 B=2↑I
120 A=B-1
130 C=B+1
140 PRINT I;C-B;B-A;B
150 IF C-B<>1 OR B-A<>1 THEN 170
160 NEXT I
170 N=I*LOG(2)/LOG(10)
180 N=INT(10*N)/10
190 PRINT:PRINT I;"BITS IN MANTISSA":PRINT
200 PRINT N;"DECIMAL DIGITS OF PRECISION":PRINT
210 TRAP 270
220 FOR I=1 TO 100
230 N=2↑I
240 X=2↑(N-2)
250 PRINT I;N-1;X
260 NEXT I
270 PRINT I-1;"BITS IN CHARACTERISTIC"
280 END
```

the present case, M,T, and U are not larger than 3 digits each for any number up to 999,999,999. While this number can be expressed without loss of precision in 9.5 digit BASIC, the product of two such numbers will have 18 digits and 8 or 9 of the least significant digits will be lost. Addition and subtraction are done by parts, with proper attention to carry (if U1 = 843 and '2 = 417, then U1 + U2 = 1260. I.e., the result has U = 260 and there is a carry of 1 into T). Multiplication follows the rule for multiplication of polynomials:

```
(M1*10↑6+T1*10↑3+U1)*(M2*10↑6+
T2*10↑3+U2) = M1*M2*10↑12+
(M1*T2+M2*T1)*10↑9+(M1*U2+M2-
*U1+T1*T2)*10↑6+(T1*U2+T2*U1)*
10↑3+U1*U2
```

Of course, the powers of ten are not explicitly entered, or you have gained nothing. Rather, the calculations are done on the coefficients, and the partial products carried along. If combining the terms at the end after multiplying by the appropriate powers of ten gives adequate precision, the results can simply be multiplied out and printed in the usual way. But if the extra precision thus gained is wanted in the result, the numbers must be converted using STR$ and the character strings which result are then concatenated to give the final result. In order to take account of carry correctly, addition, subtraction and multiplication must proceed from right to left; while division is calculated from left to right.

A practical application of this method is illustrated by the program in Listing 2. Although there are many calendar programs available, this is the only one I know of for calculating the date of the Jewish New Year. The Jewish calendar is a lunar calendar, with months of length alternately 29 and 30 days, so that the first of every month falls within a day or so of a new moon. Twelve such months total only 354 days, about 11.25 less than a solar year. In order to avoid the large errors that would arise from a deficit of 11 days per year, seven leap years of 13 months each are distributed over a 19 year cycle. The difference between 19 solar years and 235 lunar months is only 1.44907 hours. The length of the year may be adjusted by plus or minus one day in order that the New Year will not fall on Sunday, Wednesday or Friday. These small adjustments over the years also compensate for the extra 1.449 hours mentioned above. For further information and for derivation of the method and formulas used see two articles by Louis A. Resnikoff in *Scripta Mathematica* 9,191-195, 274-277(1943). The only part of the calculation that requires extended precision arithmetic is evaluation of the following division:

(31524 + (235*C + 12*m + 13*n)*765433)/181440,

where C is an integer with a value about 300 at the present time, and m and n are small integers. The quotient is not required here, but the remainder points to the day of the week on which the New Year occurs through a table. This division cannot be done without error in 9.5 digit BASIC. It is programmed in lines 2000-2330. Although the quotient is not required, as noted above, it is calculated in the subroutine for illustrative purposes.

The program is written in "standard" BASIC and should run with little or no change on most microcomputers. To use the program, simply enter the common or calendar year in response to the prompting message. The results are printed to the screen in the following form:

```
ENTER CIVIL YEAR?

1984

CIVIL YEAR = 1984

JEWISH YEAR = 5745

NEW YEAR'S DAY IS ON

THU SEP 27

ORDINARY YEAR 354 DAYS

FIRST DAY OF PASSOVER

IS ON APR 6 1985
```

The program gives not only the date of the New Year, but also tells whether the year is an ordinary year or a leap year, the exact number of days in the year, and the date of the first day of Passover.

As the accompanying listing illustrates, it is relatively direct to write programs in BASIC that will handle arithmetic calculations with any desired degree of precision, by breaking the problem into smaller parts and doing the calculations for each part separately.

The program gives not only the date of the New Year, but also tells whether the year is an ordinary year or a leap year, the exact number of days in the year, and the date of the first day of Passover.

To construct a calendar for any given year it is necessary to know the arrangement of months in the Jewish year. The transliteration of Hebrew characters is apparently not fully agreed on; I have used a scheme that seems to be widely accepted. In an ordinary year of 354 days the months are as follows: Tishri, 30 days; Heshvan, 29 days; Kislev, 30 days; Tebeth, 29 days; Shebat, 30 days; Adar, 29 days; Nisan, 30 days; Iyar, 29 days; Sivan, 30 days; Tammuz, 29 days; Ab, 30 days; and Elul, 29 days. Some of the important holidays are New Year, Tishri 1; Yom Kippur, Tishri 10; Hanukkah, Kislev 25-Tebeth 2 or 3 (see below on the length of Kislev); and Passover, Nisan 15-21. There are three possible adjustments to this calendar. If the year is a leap year, then a thirteenth month of 29 days called Adar Sheni, or Second Adar, is interpolated between Adar (now First Adar) and Nisan. First Adar is increased to 30 days. If the length of the year is 353 or 383 days (defective year), Kislev is shortened to 29 days. If the length of the year is 355 or 385 days (full year), then Heshvan is increased to 30 days.

For the current year of 1983-84 (Jewish year 5744), the program gives the date of the New Year as September 8, 1983 and the length of the year as 385 days (both a leap year and a full year). The correspondence with the civil calendar can be tabulated as follows:

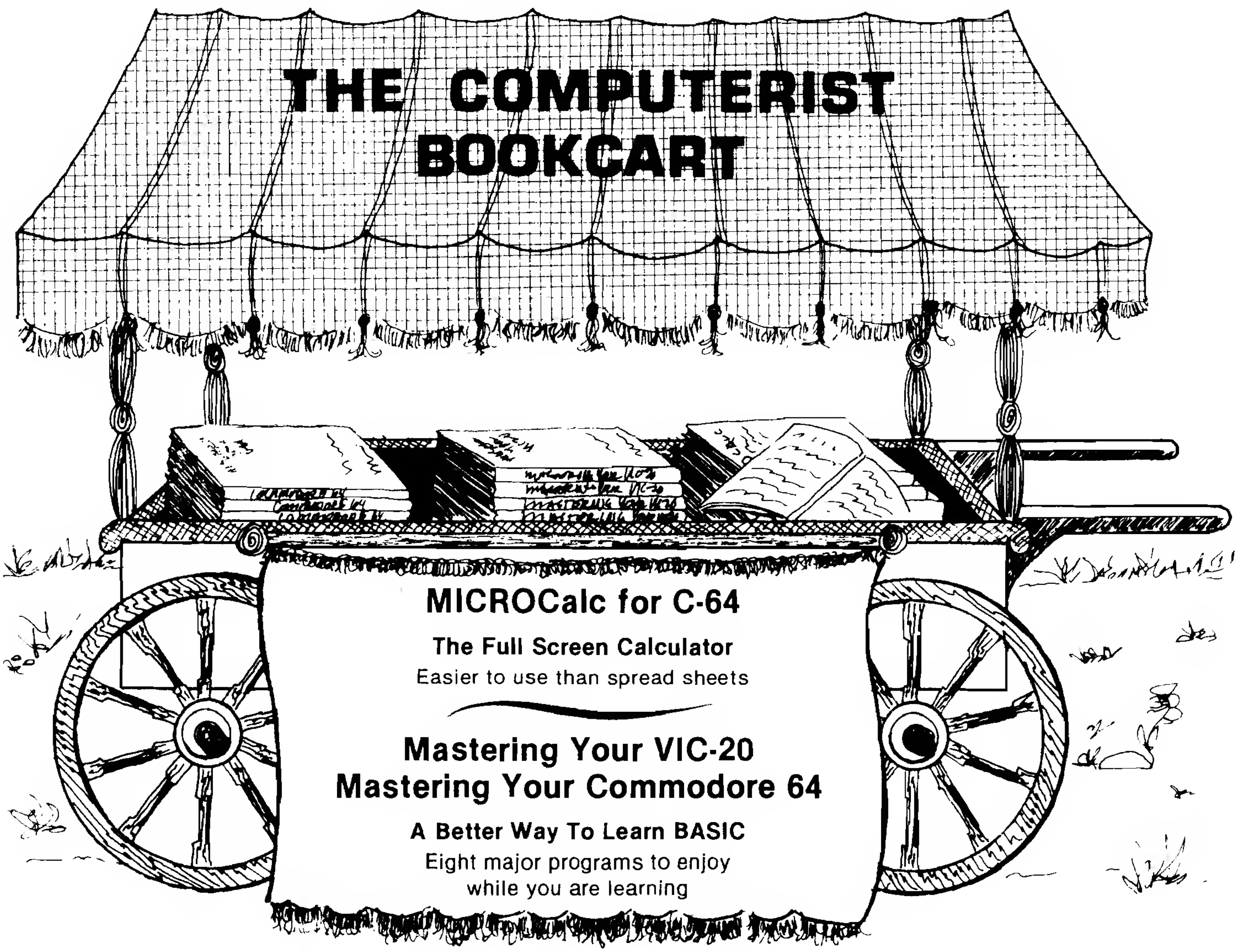
Jewish date	**Civil date**
Tishri 1, 5744	Sep 8,1983
Heshvan 1	Oct 8
Kislev 1	Nov 7
Tebeth 1	Dec 7
Shebat 1	Jan 5, 1984
First Adar 1	Feb 4
Second Adar	1Mar 5
Nisan 1	Apr 3
Iyar 1	May 3
Sivan 1	Jun 1
Tammuz 1	Jul 1
Ab 1	Jul 30
Elul 1	Aug 29
Tishri 1,5745	Sep 27

As the accompanying listing illustrates, it is relatively direct to write programs in BASIC that will handle arithmetic calculations with any desired degree of precision, by breaking the problem into smaller parts and doing the calculations for each part separately.

```
10 REM PROGRAM TO CALCULATE DATE OF JEWISH NEW YEAR
20 REM METHOD BASED ON NOTES BY L.A.RESNIKOFF
30 REM "Scripta Mathematica" 9,191-195,274-277(1943).
40 REM WRITTEN BY Rolf B. Johannesen
50 REM LAST REVISION 20 MAR 1984
60 DIM RC(7,4),RT(6,4)
65 B$=CHR$(32)
70 FOR R1=1 TO 4
80 READ RN
90 FOR R2= 1 TO RN
100 READ RC(R2,R1):NEXT R2
110 FOR R2 =1 TO 6
120 READ RT(R2,R1):NEXT R2:NEXT R1
130 PRINT "ENTER CIVIL YEAR"
140 INPUT YR
150 JY=YR+3761
160 REM FIND NO. OF 19-YEAR CYCLE
170 C=INT((JY-1)/19)
180 REM AND YEAR NO. IN THAT CYCLE
190 R=JY-19*C
230 REM SET YEAR TYPE AS ORDINARY
240 REM CHANGE LATER IF A LEAP YEAR
250 Y$="ORDINARY":YL=354
260 FOR K=1 TO 4:FOR J=1 TO 7
270 IF R=RC(J,K) THEN 290
280 NEXT J : NEXT K
290 ON K GOTO 400,400,400,300
300 REM LEAP YEAR (13 MONTHS)
310 Y$="LEAP":YL=384
320 N=INT((R-1)/3)
330 GOSUB 2010 : GOTO 420
400 REM ORDINARY YEAR (12 MONTHS)
410 GOSUB 2000
420 FOR RR=1 TO 6
430 IF FR< =RT(RR,K) THEN 500
440 NEXT RR
450 IF FR< =174959 THEN YT=1:GOTO 810
460 J=9:YT=-1:GOTO 620
500 ON RR GOTO 550,600,650,700,750,800
550 YT=-1:GOTO 610
600 YT=1
610 J=2
620 D$="MON":GOTO 1000
650 YT=0:J=3
660 D$="TUE":GOTO 1000
700 YT=-INT(K/4)
710 GOTO 760
750 YT=1
760 J=5:D$="THU"
770 GOTO 1000
800 YT=-1
810 J=7:D$="SAT"
1000 REM NOW WE HAVE DAY OF WEEK
1010 REM NEXT CALCULATE DATE
1020 Q=(-332844+1565*C+282084*M-483349*N+FR)/25920
1030 IQ=22+INT(YR/100)-INT(YR/400)-INT(Q+0.75)+J
1040 YL=YL+YT
1050 IF IQ> 30 THEN 1090
1060 M$="SEP"
1070 DT=IQ
1080 GOTO 1105
1090 M$="OCT"
1100 DT=IQ-30
1105 GOSUB 3100
1110 PRINT:PRINT:PRINT
1120 PRINT("  CIVIL YEAR = "+STR$(YR))
1130 PRINT:PRINT(" JEWISH YEAR = "+STR$(JY))
1140 PRINT:PRINT" NEW YEAR'S DAY IS ON"
1150 P$=B$+B$+D$+B$+M$+B$+STR$(DT)
1160 PRINT:PRINT P$
1170 PRINT:PRINT(Y$+B$+"YEAR"+B$+B$);
```

```
1180 PRINT (STR$(YL)+B$+"DAYS")
1182 PRINT
1185 PRINT " FIRST DAY OF PASSOVER"
1186 PRINT:PRINT (" IS ON "+PM$+STR$(PD)+STR$(PY))
1190 PRINT : PRINT : PRINT "AGAIN";
1200 INPUT A$
1210 IF ASC(A$)=89 THEN 130
1220 END
2000 N=INT(R/3)
2010 M=R-N-1
2020 REM SUBROUTINE TO FORM AN 11 DIGIT PRODUCT
2030 REM BY MULTIPLYING A 6 DIGIT * A 5 DIGIT NUMBER
2040 REM AND DIVIDING BY A 6 DIGIT NUMBER TO GET
2050 REM A QUOTIENT AND REMAINDER WITH NO LOSS OF PRECISION
2060 REM Tn = THOUSANDS; Un = UNITS; MI = MILLIONS
2070 PR=235*C+12*M+13*N
2080 T1=INT(PR/1000)
2090 U1=PR-1000*T1
2100 T2=765:U2=433
2110 MI=T1*T2
2120 TH=T1*U2+T2*U1
2130 U=U1*U2+31524
2140 U=U+1000*(TH-1000*INT(TH/1000))
2150 MI=MI+INT(TH/1000)+INT(U/1E6)
2160 U=U-1E6*INT(U/1E6)
2170 REM QUOTIENT NOT NEEDED FOR THIS PROBLEM
2180 REM BUT CARRIED THROUGH FOR ILLUSTRATION
2190 QU=0
2200 DV=181440
2210 FOR I=1 TO 6
2220 TD=INT(MI/DV)
2230 QU=10*QU+TD
2240 FR=MI-TD*DV
2250 TU=INT(U/10↑(6-I))
2260 MI=FR*10+TU
2270 U=U-TU*10↑(6-I)
2280 U=INT(U+0.5)
2290 NEXT I
2300 TD=INT(MI/DV)
2310 QU=10*QU+TD
2320 FR=MI-TD*DV
2330 RETURN
2900 REM DATA STATEMENTS DIVISIBLE BY 20 GIVE YEARS IN CYCLE
2910 REM WITH ATTRIBUTES LISTED IMMEDIATELY PRECEDING
2920 REM ALTERNATE DATA STATEMENTS GIVE NUMERATORS OF
2930 FRACTION OF A WEEK
2940 REM DENOMINATOR OF FRACTION ALWAYS 181440
2990 REM ORDINARY YEARS THROUGH 3050
2995 REM YEARS FOLLOWING AN ORDINARY YEAR NEXT
3000 DATA 5,2,5,10,13,16
3010 DATA 9923,45359,61763,113603,123119,139523
3015 REM YEARS PRECEDING AN ORDINARY YEAR NEXT
3020 DATA 5,1,4,9,12,15
3030 DATA 9923,42708,61763,113603,123119,130007
3035 REM YEARS BETWEEN TWO LEAP YEARS NEXT
3040 DATA 2,7,18
3050 DATA 9923,42708,61763,113603,123119,139523
3055 REM LEAP YEARS NEXT
3060 DATA 7,3,6,8,11,14,17,19
3070 DATA 22090,45359,71279,90334,123119,151690
3100 SU=344:PY=YR+1
3110 IF 400*INT(PY/400)=PY THEN 3140
3120 IF 100*INT(PY/100)=PY THEN 3150
3130 IF 4*INT(PY/4) < PY THEN 3150
3140 SU=345
3150 PD=YL+IQ-SU
3160 IF PD> 31 THEN 3190
3170 PM$="MAR"
3180 RETURN
3190 PM$="APR":PD=PD-31
3200 RETURN
```

MICRO



Mastering Your VIC-20
Mastering Your Commodore 64

The 8 programs, "run-ready" on disk (C-64) or tape (VIC-20) and explained in the 160-192 page book, each demonstrate important concepts of BASIC while providing useful, enjoyable software. Programs include:

- Player — compose songs from your keyboard, save, load and edit for perfect music
- MicroCalc — display calculation program that make even complex operations easy
- Master — a one or two person guessing game
- Clock — character graphics for a digital clock

VIC-20 with tape & book .... **just $19.95**
C-64 with disk & book (avail. Sept.) ..... **just $19.95**

Look for us at the
International Software Show
Toronto, September 20-23

MICROCalc for C-64

This on-screen calculator comes with diskette and 48-page manual offering a wide variety of useful screens, and a great way to learn BASIC expressions if you don't already know them.

- Unlimited calculation length & complexity
- Screens can be linked and saved on disk/cassette
- Build a library of customized screens
- Provide formatted printer output

Diskette & 48-page manual .... **just $29.95**

For the Freshest Books, Buy Direct!

- No prehandled books with bent corners
- Books come direct to your door
- No time wasted searching store to store
- 24 hours from order receipt to shipment
- No shipping/handling charges
- No sales tax (except 5% MA res.)
- Check, MO, VISA/MC accepted (prepaid only)

The Computerist Bookcart
P.O. Box 6502, Chelmsford, MA 01824
For faster service, phone: 617/256 – 3649.

HILISTER - *A Study and Teaching Aid*

(Part 2)

by J. Morris Prosser

Move easily within your programs and highlight parts of text or listings for emphasis or clarity.

Part 1 of this article described the HILISTER program and included the highlighter portion of the assembly listing. This part will include the remainder of the listing and will explain how to interface it to the first part.

If you have an assembler, the source code for the first part should be loaded, then the second part should be added. You may prefer to enter the second part as a separate source file, then append it to the first part. If you do that, you will not be able to assemble it separately, since it is not complete in itself. In either case, UPDO and DOWNDO should be removed from the list of equates, since they are internal labels in the second part of the source. At label LISTER, change the operation code mnemonic from RTS to CLD, so that execution falls through to the second part of the program. Once the code has been entered and these changes have been made, the code can be assembled and the source and object code saved as HILISTER. The program should now be complete.

If you are entering the code without an assembler, it is best to enter the second part of the code separately. It should then be saved:

BSAVE HILISTER2,A$80D0,L$540

In the process of entering the code, it is a good idea to stop every now and then (say every screenful) and save what you have entered to disk, using the same command as shown above. If you now BLOAD HILISTER1 and BLOAD HILISTER2, you will be ready to make the changes necessary to integrate the two sections. Go to the monitor (CALL -151) and enter 8062.8063, remembering that a carriage return is required after each entry. You should see 8062- 58 FF if the program has been entered correctly. Now enter 8062:83 83. Next enter 8087.8088. You should see 8087- 58 FF. Enter 8087:65 83. One more change is needed - enter 80CF, and you should see 80CF- 60. Enter 80CF:D8, and you are finished. Save the program back to your disk:

BSAVE HILISTER,A$8000,L$610

You should be able to use the program now by entering BRUN HILISTER.

NOTE: In last month's listing of hilister (part I) line 8008 should read

```
8ØØ8 A9 1B    LDA BEGIN
```

line 8010 should read

```
8Ø1Ø A9 8Ø    LDA BEGIN
```

```
          * Determine type of listing and
          * branch accordingly. If A =
          * #188 = Applesoft
          * $    = monitor
          * C    = Catalog
          * B    = setup to Begin
          * E    = End of listing
          * anything else is error
80D0 C9 BC                CMP #188
80D2 D0 08                BNE NOTLIST
80D4 85 1A                STA LSTFLG
80D6 20 0A 81             JSR OUTSET
80D9 4C 9D 81             JMP LISTST
80DC C9 24      NOTLIST   CMP #'$'
80DE D0 08                BNE NOTLIST1
80E0 85 F9                STA MEMFLG
80E2 20 0A 81             JSR OUTSET
80E5 4C 4B 84             JMP MEMLST
80E8 C9 43      NOTLIST1  CMP #'C'
80EA D0 08                BNE NOTC
80EC 85 1D                STA CATFLG
80EE 20 0A 81             JSR OUTSET
80F1 4C 76 85             JMP CTLG
80F4 C9 42      NOTC      CMP #'B'
80F6 D0 06                BNE NOTB
80F8 20 0A 81             JSR OUTSET
80FB 4C D0 03             JMP BASIC
80FE C9 45      NOTB      CMP #'E'
8100 D0 05                BNE ERROR
8102 85 1A                STA LSTFLG
8104 4C 2F 82             JMP ENDLST
8107 4C C9 DE   ERROR     JMP SYNERR
          *  Detour character output,
          *  TELLDOS to make DOS happy, and
          *  Set/test buffer pointers
810A A0 56      OUTSET    LDY #< OUTST
810C 84 36                STY CSWL
810E A0 81                LDY #> OUTST
8110 84 37                STY CSWL+1
8112 20 EA 03             JSR TELLDOS
8115 A0 00                LDY #< BUFLE
8117 84 FA                STY BUFST
8119 A0 40                LDY #> BUFLE
811B 84 FB                STY BUFST+1
811D A4 1A                LDY LSTFLG
811F F0 09                BEQ BUFOK
8121 A4 B0                LDY $B0
8123 C8                   INY
8124 C4 FB                CPY BUFST+1
8126 90 02                BCC BUFOK
8128 84 FB                STY BUFST+1
          * Set up to fill buffer with carriage returns
          * Set screen start and end of listing area
812A A4 FA      BUFOK     LDY BUFST
812C 84 3C                STY A1L
812E 84 FC                STY SCRST
8130 C8                   INY
8131 84 42                STY A4L
8133 A4 FB                LDY BUFST+1
8135 84 3D                STY A1L+1
8137 84 43                STY A4L+1
8139 84 FD                STY SCRST+1
813B 38                   SEC
813C A9 FF                LDA #< START-1
813E 85 3E                STA A2L
8140 E9 18                SBC #24
8142 85 FE                STA LSTEND
8144 A9 7F                LDA #> START-1
8146 85 3F                STA A2L+1
8148 E9 00                SBC #0
814A 85 FF                STA LSTEND+1
814C A9 8D                LDA #$8D
814E A0 00                LDY #0
8150 91 FA                STA (BUFST),Y
8152 20 2C FE             JSR MOVE
8155 60                   RTS
          * Character output detour
          * Print to screen and update
          * and test buffer pointers
8156 84 07      OUTST     STY TEMPY
8158 20 F0 FD             JSR COUT1
815B A0 00                LDY #0
815D 91 FC                STA (SCRST),Y
815F 20 D8 83             JSR NXTLOC
8162 B0 03                BCS TOOLONG
8164 A4 07                LDY TEMPY
8166 60                   RTS
          * Too Long - ring bell and show last lines
          * Get keyboard to determine what to do next
8167 20 3A FF   TOOLONG   JSR BELL
816A 20 83 82             JSR PRTSCRN
816D A0 00                LDY #0
816F B9 A0 85   MSGLP     LDA TOOLNG,Y
8172 F0 06                BEQ GETREQ
8174 20 F0 FD             JSR COUT1
8177 C8                   INY
8178 D0 F5                BNE MSGLP
817A 2C 00 C0   GETREQ    BIT KBD
817D 10 FB                BPL GETREQ
817F AD 00 C0             LDA KBD
8182 2C 10 C0             BIT KBDSTRB
8185 C9 8D                CMP #$8D
8187 D0 09                BNE ESCCHK
8189 20 83 82             JSR PRTSCRN
```

```
818C 20 03 84            JSR SCRNPRT
818F 4C 9A 82            JMP GETCMD
8192 C9 9B      ESCCHK   CMP #$9B
8194 D0 E4               BNE GETREQ
8196 20 58 FC            JSR HOME
8199 4C D0 03            JMP BASIC
              *
              * Applesoft list
              *
819C 60         LRTN     RTS
819D A2 FF      LISTST   LDX #$FF
819F 20 B1 00            JSR CHRGET
              *
              * Replacement for Applesoft LIST routine
              *
81A2 90 0A      LIST     BCC LIST1
81A4 F0 08               BEQ LIST1
81A6 C9 C9               CMP #$C9
81A8 F0 04               BEQ LIST1
81AA C9 2C               CMP #$2C
81AC D0 EE               BNE LRTN
81AE 20 0C DA   LIST1    JSR LINGET
81B1 20 1A D6            JSR FNDLIN
81B4 20 B7 00            JSR CHRGOT
81B7 F0 10               BEQ LIST3
81B9 C9 C9               CMP #$C9
81BB F0 04               BEQ LIST2
81BD C9 2C               CMP #$2C
81BF D0 DB               BNE LRTN
81C1 20 B1 00   LIST2    JSR CHRGET
81C4 20 0C DA            JSR LINGET
81C7 D0 D3               BNE LRTN
81C9 68         LIST3    PLA
81CA 68                  PLA
81CB A5 50               LDA LINNUM
81CD 05 51               ORA LINNUM+1
81CF D0 06               BNE LIST4
81D1 A9 FF               LDA #$FF
81D3 85 50               STA LINNUM
81D5 85 51               STA LINNUM+1
81D7 A0 01      LIST4    LDY #$01
81D9 B1 9B               LDA (LOWTR),Y
81DB F0 4D               BEQ LIST10
81DD 20 D0 84            JSR ISCNTC
81E0 20 FB DA            JSR CRDO
81E3 C8                  INY
81E4 B1 9B               LDA (LOWTR),Y
81E6 AA                  TAX
81E7 C8                  INY
81E8 B1 9B               LDA (LOWTR),Y
81EA C5 51               CMP LINNUM+1
81EC D0 04               BNE LIST5
81EE E4 50               CPX LINNUM
81F0 F0 02               BEQ LIST6
81F2 B0 36      LIST5    BCS LIST10
81F4 84 85      LIST6    STY FORPNT
81F6 20 24 ED            JSR LINPRT
81F9 A9 20               LDA #$20
81FB A4 85      LIST7    LDY FORPNT
81FD 29 7F               AND #$7F
81FF 20 5C DB   LIST8    JSR OUTDO
8202 A5 24               LDA CH
8204 C9 21               CMP #$21
8206 90 10               BCC LIST9
8208 20 FB DA            JSR CRDO
820B A9 05               LDA #5
820D 85 1B               STA COUNT
820F A9 A0      LSTLOOP  LDA #$A0
8211 20 5C DB            JSR OUTDO
8214 C6 1B               DEC COUNT
8216 D0 F7               BNE LSTLOOP
8218 C8         LIST9    INY
8219 B1 9B               LDA (LOWTR),Y
821B D0 33               BNE LIST13
821D A8                  TAY
821E B1 9B               LDA (LOWTR),Y
8220 AA                  TAX
8221 C8                  INY
8222 B1 9B               LDA (LOWTR),Y
8224 86 9B               STX LOWTR
8226 85 9C               STA LOWTR+1
8228 D0 AD               BNE LIST4
822A A9 0D      LIST10   LDA #$0D
822C 20 5C DB            JSR OUTDO
              *
              * End of List processing
              *
822F 20 83 82   ENDLST   JSR PRTSCRN
8232 20 03 84            JSR SCRNPRT
8235 20 3A FF            JSR BELL
8238 A0 00               LDY #0
823A B9 EC 85   MSGLP1   LDA ENDMSG,Y
823D F0 06               BEQ LISTEND
823F 20 F0 FD            JSR COUT1
8242 C8                  INY
8243 D0 F5               BNE MSGLP1
8245 4C 9A 82   LISTEND  JMP GETCMD
8248 C8         LIST11   INY
8249 D0 02               BNE LIST12
824B E6 9E               INC DSCTMP+1
824D B1 9D      LIST12   LDA (DSCTMP),Y
824F 60                  RTS
8250 10 AD      LIST13   BPL LIST8
8252 38                  SEC
8253 E9 7F               SBC #$7F
8255 AA                  TAX
8256 84 85               STY FORPNT
8258 A0 D0               LDY #$D0
825A 84 9D               STY DSCTMP
825C A0 CF               LDY #$CF
825E 84 9E               STY DSCTMP+1
8260 A0 FF               LDY #$FF
8262 CA         LIST14   DEX
8263 F0 07               BEQ LIST16
8265 20 48 82   LIST15   JSR LIST11
8268 10 FB               BPL LIST15
826A 30 F6               BMI LIST14
826C A9 20      LIST16   LDA #$20
826E 20 5C DB            JSR OUTDO
8271 20 48 82   LIST17   JSR LIST11
8274 30 05               BMI LIST18
8276 20 5C DB            JSR OUTDO
8279 D0 F6               BNE LIST17
827B 20 5C DB   LIST18   JSR OUTDO
827E A9 20               LDA #$20
8280 4C FB 81            JMP LIST7
              *
              * Set up to print last lines of
              * listing to screen
8283 20 BC 83   PRTSCRN  JSR PGBAK
8286 A5 FC               LDA SCRST
8288 85 FE               STA LSTEND
828A A5 FD               LDA SCRST+1
828C 85 FF               STA LSTEND+1
828E A9 F0               LDA #<COUT1
8290 85 36               STA CSWL
8292 A9 FD               LDA #>COUT1
8294 85 37               STA CSWL+1
8296 20 EA 03            JSR TELLDOS
8299 60                  RTS
```

```
        *
        *  Get keyboard command
        *
829A 2C 00 C0   GETCMD   BIT KBD
829D 10 FB               BPL GETCMD
829F AD 00 C0            LDA KBD
82A2 2C 10 C0            BIT KBDSTRB
82A5 C9 A0               CMP #$A0
82A7 D0 10               BNE NOTSPC
82A9 A5 1E               LDA DIRFLG
82AB 30 06               BMI DOWN
82AD 20 65 83            JSR UPDO
82B0 4C 9A 82            JMP GETCMD
82B3 20 83 83   DOWN     JSR DOWNDO
82B6 4C 9A 82            JMP GETCMD
82B9 C9 88      NOTSPC   CMP #$88
82BB D0 06               BNE NOTDN
82BD 20 4E 83            JSR SCRLDN
82C0 4C 9A 82            JMP GETCMD
82C3 C9 95      NOTDN    CMP #$95
82C5 D0 06               BNE NOTUP
82C7 20 37 83            JSR SCRLUP
82CA 4C 9A 82            JMP GETCMD
82CD C9 9B      NOTUP    CMP #$9B
82CF D0 03               BNE NOTOUT
82D1 4C D0 03            JMP BASIC
82D4 C9 AB      NOTOUT   CMP #$AB
82D6 D0 06               BNE NOTAB
82D8 20 A0 83            JSR PGFWD
82DB 4C 9A 82            JMP GETCMD
82DE C9 BB      NOTAB    CMP #$BB
82E0 D0 06               BNE NOTBB
82E2 20 A0 83            JSR PGFWD
82E5 4C 9A 82            JMP GETCMD
82E8 C9 AD      NOTBB    CMP #$AD
82EA D0 06               BNE NOTAD
82EC 20 BC 83            JSR PGBAK
82EF 4C 9A 82            JMP GETCMD
82F2 C9 BD      NOTAD    CMP #$BD
82F4 D0 06               BNE NOTBD
82F6 20 BC 83            JSR PGBAK
82F9 4C 9A 82            JMP GETCMD
82FC C9 A6      NOTBD    CMP #$A6
82FE D0 06               BNE NOTA6
8300 20 2E 80            JSR HILITER1
8303 4C 9A 82            JMP GETCMD
8306 C9 C2      NOTA6    CMP #$C2
8308 D0 0E               BNE NOTC2
830A A5 FA               LDA BUFST
830C 85 FC               STA SCRST
830E A5 FB               LDA BUFST+1
8310 85 FD               STA SCRST+1
8312 20 03 84            JSR SCRNPRT
8315 4C 9A 82            JMP GETCMD
8318 C9 C5      NOTC2    CMP #$C5
831A F0 03               BEQ C5
831C 4C 9A 82            JMP GETCMD
831F A5 FE      C5       LDA LSTEND
8321 85 FC               STA SCRST
8323 A5 FF               LDA LSTEND+1
8325 85 FD               STA SCRST+1
8327 20 03 84            JSR SCRNPRT
832A 4C 9A 82            JMP GETCMD
        *
        * Look for keypress to stop scroll
        *
832D 2C 00 C0   KBDCHK   BIT KBD
8330 10 04               BPL NOKEY
8332 2C 10 C0            BIT KBDSTRB
8335 38                  SEC
8336 60         NOKEY    RTS
        *
        * Scroll up routine
        *
8337 20 D8 83   SCRLUP   JSR NXTLOC
833A 90 09               BCC SCRLUP2
833C 20 3A FF            JSR BELL
833F 60                  RTS
8340 20 D8 83   SCRLUP1  JSR NXTLOC
8343 B0 08               BCS UPRTN
8345 20 6E 83   SCRLUP2  JSR UPOK
8348 20 2D 83            JSR KBDCHK
834B 90 F3               BCC SCRLUP1
834D 60         UPRTN    RTS
        *
        * Scroll down routine
        *
834E 20 EA 83   SCRLDN   JSR LSTLOC
8351 90 09               BCC SCRLDN2
8353 20 3A FF            JSR BELL
8356 60                  RTS
8357 20 EA 83   SCRLDN1  JSR LSTLOC
835A B0 08               BCS DNRTN
835C 20 8C 83   SCRLDN2  JSR DWNOK
835F 20 2D 83            JSR KBDCHK
8362 90 F3               BCC SCRLDN1
8364 60         DNRTN    RTS
        *
        * Scroll up one line
        *
8365 20 D8 83   UPDO     JSR NXTLOC
8368 90 04               BCC UPOK
836A 20 3A FF            JSR BELL
836D 60                  RTS
836E A0 00      UPOK     LDY #0
8370 84 1E               STY DIRFLG
8372 20 D8 83            JSR NXTLOC
8375 B1 FC      UPLOOP   LDA (SCRST),Y
8377 C9 8D               CMP #$8D
8379 F0 05               BEQ UPDONE
837B 20 D8 83            JSR NXTLOC
837E 90 F5               BCC UPLOOP
8380 4C 03 84   UPDONE   JMP SCRNPRT
        *
        * Scroll down one line
        *
8383 20 EA 83   DOWNDO   JSR LSTLOC
8386 90 04               BCC DWNOK
8388 20 3A FF            JSR BELL
838B 60                  RTS
838C A0 00      DWNOK    LDY #0
838E 84 1E               STY DIRFLG
8390 C6 1E               DEC DIRFLG
8392 B1 FC      DWNLOOP  LDA (SCRST),Y
8394 C9 8D               CMP #$8D
8396 F0 05               BEQ DWNDONE
8398 20 EA 83            JSR LSTLOC
839B 90 F5               BCC DWNLOOP
839D 4C 03 84   DWNDONE  JMP SCRNPRT
        *
        * Page forward
        *
83A0 20 D8 83   PGFWD    JSR NXTLOC
83A3 90 04               BCC PGFOK
83A5 20 3A FF            JSR BELL
83A8 60                  RTS
83A9 A2 17      PGFOK    LDX #23
83AB A0 00               LDY #0
83AD 20 D8 83   PGFLOOP  JSR NXTLOC
83B0 B1 FC               LDA (SCRST),Y
```

```
83B2 C9 8D              CMP #$8D
83B4 DØ F7              BNE PGFLOOP
83B6 CA                 DEX
83B7 DØ F4              BNE PGFLOOP
83B9 4C Ø3 84           JMP SCRNPRT
            *
            * Page backward
            *
83BC 2Ø EA 83  PGBAK    JSR LSTLOC
83BF 9Ø Ø4              BCC PGBOK
83C1 2Ø 3A FF           JSR BELL
83C4 6Ø                 RTS
83C5 A2 17     PGBOK    LDX #23
83C7 AØ ØØ              LDY #Ø
83C9 2Ø EA 83  PGBLOOP  JSR LSTLOC
83CC B1 FC              LDA (SCRST),Y
83CE C9 8D              CMP #$8D
83DØ DØ F7              BNE PGBLOOP
83D2 CA                 DEX
83D3 DØ F4              BNE PGBLOOP
83D5 4C Ø3 84           JMP SCRNPRT
            *
            * Increment screen pointer
            *
83D8 A5 FC     NXTLOC   LDA SCRST
83DA C5 FE              CMP LSTEND
83DC A5 FD              LDA SCRST+1
83DE E5 FF              SBC LSTEND+1
83EØ 9Ø Ø1              BCC NXTINC
83E2 6Ø                 RTS
83E3 E6 FC     NXTINC   INC SCRST
83E5 DØ Ø2              BNE NXT1
83E7 E6 FD              INC SCRST+1
83E9 6Ø        NXT1     RTS
            *
            * Decrement screen pointer
            *
83EA A5 FA     LSTLOC   LDA BUFST
83EC C5 FC              CMP SCRST
83EE A5 FB              LDA BUFST+1
83FØ E5 FD              SBC SCRST+1
83F2 9Ø Ø1              BCC LSTDEC
83F4 6Ø                 RTS
83F5 C6 FC     LSTDEC   DEC SCRST
83F7 A4 FC              LDY SCRST
83F9 CØ FF              CPY #$FF
83FB DØ Ø2              BNE LSTRTN
83FD C6 FD              DEC SCRST+1
83FF 18        LSTRTN   CLC
84ØØ AØ ØØ              LDY #Ø
84Ø2 6Ø                 RTS
            *
            * Screen print routine
            *
84Ø3 A5 FC     SCRNPRT  LDA SCRST
84Ø5 85 3C              STA A1L
84Ø7 A5 FD              LDA SCRST+1
84Ø9 85 3D              STA A1L+1
84ØB A9 ØØ              LDA #Ø
84ØD 85 24              STA CH
84ØF 85 25              STA CV
8411 2Ø 22 FC           JSR VTAB
8414 A2 18              LDX #24
8416 AØ ØØ     LOOP2    LDY #Ø
8418 2Ø BA FC           JSR NXTA1
841B B1 3C              LDA (A1L),Y
841D C9 8D              CMP #$8D
841F DØ ØA              BNE PRNT
8421 48                 PHA
8422 2Ø 9C FC           JSR CLREOL
8425 68                 PLA
8426 CA                 DEX
8427 DØ 1C              BNE PRNT1
8429 18                 CLC
842A 6Ø                 RTS
842B C9 8A     PRNT     CMP #$8A
842D DØ 16              BNE PRNT1
842F A5 24              LDA CH
8431 48                 PHA
8432 2Ø 9C FC           JSR CLREOL
8435 A9 8D              LDA #$8D
8437 2Ø FØ FD           JSR COUT1
843A 2Ø 9C FC           JSR CLREOL
843D 68                 PLA
843E 85 24              STA CH
844Ø CA                 DEX
8441 DØ D3              BNE LOOP2
8443 18                 CLC
8444 6Ø                 RTS
8445 2Ø FØ FD  PRNT1    JSR COUT1
8448 4C 16 84           JMP LOOP2
            *
            * Check for $B8 (A/S token for "DEF")
            * If found, replace with "DEF"
            *
844B AØ FF     MEMLST   LDY #$FF
844D C8        NEXT     INY
844E B9 ØØ Ø2           LDA IN,Y
8451 FØ 2F              BEQ B8OK
8453 C9 B8              CMP #$B8
8455 DØ F6              BNE NEXT
8457 C8        NEXT1    INY
8458 B9 ØØ Ø2           LDA IN,Y
845B DØ FA              BNE NEXT1
845D 99 Ø2 Ø2           STA IN+2,Y
846Ø 99 Ø4 Ø2           STA IN+4,Y
8463 88        NEXT2    DEY
8464 B9 ØØ Ø2           LDA IN,Y
8467 99 Ø2 Ø2           STA IN+2,Y
846A C9 B8              CMP #$B8
846C DØ F5              BNE NEXT2
846E A9 44              LDA #$44
847Ø 99 ØØ Ø2           STA IN,Y
8473 C8                 INY
8474 A9 45              LDA #$45
8476 99 ØØ Ø2           STA IN,Y
8479 C8                 INY
847A A9 46              LDA #$46
847C 99 ØØ Ø2           STA IN,Y
847F 4C 4B 84           JMP MEMLST
8482 AØ FF     B8OK     LDY #$FF
8484 C8        BACK     INY
8485 2Ø B1 ØØ           JSR CHRGET
8488 FØ Ø2              BEQ DONE
848A Ø9 8Ø              ORA #$8Ø
848C 99 ØØ Ø2  DONE     STA IN,Y
848F DØ F3              BNE BACK
8491 2Ø C7 FF           JSR ZMODE
8494 2Ø A7 FF           JSR GETNUM
8497 C9 A7              CMP #$A7
8499 DØ ØB              BNE NOTA7
849B 85 31              STA MODE
849D 2Ø A7 FF           JSR GETNUM
84AØ 2Ø F2 84           JSR DUMP
84A3 4C 2F 82           JMP ENDLST
84A6 C9 39     NOTA7    CMP #$39
84A8 FØ ØA              BEQ DISASM
84AA C9 Ø5              CMP #$Ø5
84AC FØ Ø6              BEQ DISASM
84AE C9 Ø1              CMP #$Ø1
```

```
84B0 D0 1B              BNE SNTX
84B2 85 1C              STA PLUSFLG
84B4 A5 3E     DISASM   LDA A2L
84B6 85 3A              STA PCL
84B8 A5 3F              LDA A2L+1
84BA 85 3B              STA PCL+1
84BC A9 00              LDA #0
84BE 20 DD 84           JSR MONLIST
84C1 A5 1C              LDA PLUSFLG
84C3 F0 05              BEQ NEXT3
84C5 A9 00              LDA #0
84C7 20 DD 84           JSR MONLIST
84CA 4C 2F 82  NEXT3    JMP ENDLST
84CD 4C C9 DE  SNTX     JMP SYNERR
         *
         * Replacement for A/S cntrl-C check
         *
84D0 AD 00 C0  ISCNTC   LDA KBD
84D3 C9 83              CMP #$83
84D5 D0 05              BNE CNTCRTN
84D7 68                 PLA
84D8 68                 PLA
84D9 4C 2F 82           JMP ENDLST
84DC 60        CNTCRTN  RTS
         *
         * Replacement for monitor LIST2 routine
         *
84DD 48        MONLIST  PHA
84DE 20 D0 F8           JSR INSTDSP
84E1 20 53 F9           JSR PCADJ
84E4 85 3A              STA PCL
84E6 84 3B              STY PCL+1
84E8 20 D0 84           JSR ISCNTC
84EB 68                 PLA
84EC 38                 SEC
84ED E9 01              SBC #1
84EF D0 E6              BNE MONLIST
84F1 60                 RTS
         *
         * Replacement for monitor XAM routine
         *
84F2 20 C7 FF  DUMP     JSR ZMODE
84F5 88                 DEY
84F6 D0 04              BNE FIXA3
84F8 A5 3C              LDA A1L
84FA 09 07              ORA #$07
84FC 85 3E              STA A2L
84FE A5 3D              LDA A1L+1
8500 85 3F              STA A2L+1
8502 A4 3D     FIXA3    LDY A1L+1
8504 A6 3C              LDX A1L
8506 86 40              STX A3L
8508 84 41              STY A3L+1
850A 20 CA 84           JSR ISCNTC
850D A9 8D              LDA #$8D
850F 20 ED FD           JSR COUT
8512 20 40 F9           JSR PRNTYX
8515 A0 00              LDY #$00
8517 A9 AD              LDA #$AD
8519 20 ED FD           JSR COUT
851C A9 A0     HEXOUT   LDA #$A0
851E 20 ED FD           JSR COUT
8521 B1 3C              LDA (A1L),Y
8523 20 DA FD           JSR PRBYTE
8526 20 BA FC           JSR NXTA1
8529 B0 06              BCS ASCOUT
852B A5 3C              LDA A1L
852D 29 07              AND #$07
852F D0 E5              BNE HEXOUT
8531 20 31 85           JSR ASCOUT
8534 4C FC 84           JMP FIXA3
8537 A9 A0     ASCOUT   LDA #$A0
8539 20 ED FD           JSR COUT
853C A9 1E              LDA #$1E
853E C5 24              CMP $24
8540 B0 EF              BCS ASCOUT
8542 A0 00              LDY #$00
8544 A9 07              LDA #$07
8546 25 40              AND A3L
8548 49 FF              EOR #$FF
854A 18                 CLC
854B 69 08              ADC #$08
854D 48        GETNXT   PHA
854E B1 40              LDA (A3L),Y
8550 29 7F              AND #$7F
8552 C9 20              CMP #$20
8554 90 FE              BCC SPCOUT
8556 09 80              ORA #$80
8558 D0 FC              BNE PRINT
855A A9 A0     SPCOUT   LDA #$A0
855C 20 ED FD  PRINT    JSR COUT
855F A5 40              LDA A3L
8561 C5 3E              CMP A2L
8563 A5 41              LDA A3L+1
8565 E5 3F              SBC A2L+1
8567 E6 40              INC A3L
8569 D0 FC              BNE INCEND
856B E6 41              INC A3L+1
856D 68        INCEND   PLA
856E B0 FF              BCS DMPEND
8570 38                 SEC
8571 E9 01              SBC #$01
8573 10 D2              BPL GETNXT
8575 60        DMPEND   RTS
         *
         * Catalog list routine
         *
8576 20 B1 00  CTLG     JSR CHRGET
8579 F0 05              BEQ DEFDRV
857B C9 31              CMP #$31
857D F0 FE              BEQ PUTDRV
857F C9 32              CMP #$32
8581 D0 14              BNE SNTX1
8583 8D 68 AA  PUTDRV   STA $AA68
8586 AD 34 AE  DEFDRV   LDA $AE34
8589 85 1D              STA CATFLG
858B A9 60              LDA #$60
858D 8D 34 AE           STA $AE34
8590 A9 06              LDA #06
8592 20 AA A2           JSR $A2AA
8595 A5 1D              LDA CATFLG
8597 8D 34 AE           STA $AE34
859A 4C 2F 82           JMP ENDLST
859D 4C C9 DE  SNTX1    JMP SYNERR
         *
         * Screen messages follow
         *
85A0 8D 8D     TOOLNG   HEX 8D8D
85A2 D0 D2 CF           ASC "PROGRAM TOO LONG
                             TO FIT IN BUFFER."
85C4 8D 8D              HEX 8D8D
85C6 D0 D2 C5           ASC "PRESS <RETURN> TO
                             LIST, <ESC> TO EXIT"
85EB 00                 HEX 00
         *
85EC 8D 8D     ENDMSG   HEX 8D8D
85EE A0 A0 A0           ASC "  ***  LISTING
                             COMPLETED  ***"
860E 8D 00              HEX 8D00
                        END
```

feature

USEFUL FUNCTIONS—Part 3

by Paul Garrison

In last month's issue we printed the second of three programs that allowed you to easily access various defined functions. This saved aggravation and time when working with complicated mathematical formulas. We present, as a continuation of this, the third program which will put even more valuable formulas and functions at your fingertips. Again, we invite you to send in any defined functions you may be using that are not mentioned. The submissions we receive will be published in a future issue.

Save time and aggrevation with a collection of defined functions.

```
1 REM FUNCTIONS (DELETE THOSE NOT USED IN A PROGRAM)
2 PI=3.14159
3 RAD=57.2958
4 DEF FNL(A,B)=-(A< =B)*A-(B< A)*B:                    REM LESSER  OF A AND B
5 DEF FNG(A,B)=-(A> =B)*A-(B> A)*B:                    REM GREATER OF A AND B
6 DEF FNAV(A,B)=(A+B)/2:REM AVERAGE OF A AND B
7 DEF FNDX(A,X)=INT(A*X+.5)/X:REM LIMIT TO X DECIMALS
8 DEF FNPRX(RHO,THETA)=RHO*SIN(THETA/RAD):REM P TO R, FINDS X
9 DEF FNPRY(RHO,THETA)=RHO*COS(THETA/RAD):REM P TO R, FINDS Y
1Ø DEF FNRPR(X,Y)=SQR(X↑2+Y↑2):REM R TO P, FINDS RHO
11 DEF FNRPT(X,Y)=-(X=Ø AND Y< Ø)*18Ø-(X> Ø)*(9Ø-RAD*ATN(Y/X))-(X< Ø)*(27Ø-RAD*ATN(Y/X))
12 REM R TO P, FINDS THETA
13 DEF FNSSS(A,B,C)=RAD*2*ATN(SQR(((A+B+C)/2-A)*((A+B+C)/2-B)*((A+B+C)/2-C)*2/(A+B+C))/((A+B+C)/2-A))
14 REM FINDS ANGLE OPPOSITE SIDE A,GIVEN 3 SIDES OF A TRIANGLE
15 DEF FNROOT(X,Z)=X↑(1/Z):REM Z-ROOT OF X
35 DEF FNREC(A)=1/A:&REM RECIPROCAL OF A
36 DEF FNDEG(A)=A*(PI/18Ø):REM DEGREES TO RADIANS
37 DEF FNRAD(A)=A/(PI/18Ø):REM RADIANS TO DEGREES
38 DEF FNVOLC(S)=S↑3:%REM VOLUME OF A CUBE
39 DEF FNVOLR(L,W,H)=L*W*H:REM VOLUME OF BOX
4Ø DEF FNVOLS(R)=4/3*PI*R↑3:REM VOLUME OF A SPHERE
41 DEF FNVOLP(B,H)=B*H/3:!REM VOLUME OF A PYRAMID
42 DEF FNVOLL(R,H)=PI*R↑2*H:REM VOLUME OF A CYLINDER
43 DEF FNSURC(S)=6*(S*S): REM SURFACE OF A CUBE
44 DEF FNSURR(L,W,H)=2*(L*W)+2*(L*H)+2*(W*H): REM SURFACE OF A BOX
45 DEF FNSURS(R)=4*PI*R↑2: REM SURFACE OF A SPHERE
46 DEF FNSURL(R,H)=2*PI*R↑2+2*PI*R*H:REM SURFACE OF A CYLINDER

1ØØ REM (PROGRAM TITLE, AUTHOR)
11Ø REM (TYPE OF BASIC USED)
12Ø GOTO 18Ø
13Ø ?"————————————————————————————————————":RETURN
14Ø HOME:VTAB(1Ø):RETURN
15Ø ?:INPUT "Press >RETURN< (Q to quit)",R$
```

```
155 IF R$="Q" THEN 160 ELSE RETURN
160 GOSUB 140:GOSUB 130:?TAB(33)"End.":GOSUB 130:END
180 GOSUB 140:GOTO 2000
190 REM TESTING FUNCTIONS
200 ?"Find the lesser of two numbers":GOSUB 130
210 INPUT "Enter any two numbers",A,B
220 X=FNL(A,B):GOSUB 130
230 PRINT "The lesser number is";X:GOSUB 150:GOTO 2000
240 ?"Find the greater of two numbers":GOSUB 130
250 INPUT "Enter any two numbers",A,B
260 X=FNG(A,B):GOSUB 130
270 PRINT "The greater number is",X:GOSUB 150:GOTO 2000
280 ?"Find the average of two numbers":GOSUB 130
290 INPUT "Enter any two numbers",A,B
300 X=FNAV(A,B):GOSUB 130
310 PRINT "The average of the two numbers is",X:GOSUB 150:GOTO 2000
320 ?"Round number to X decimals":GOSUB 130
330 INPUT "Enter a number with many decimals",A
340 PRINT "Enter 10, 100, 1000 etc. to limit the number of decimals"
345 INPUT "to the number of zeros",LD
350 X=FNDX(A,LD):GOSUB 130
360 PRINT "The rounded-off number is ";X:GOSUB 150:GOTO 2000
370 ?"Polar to rectangular conversion, find X":GOSUB 130
380 INPUT "Diagonal length (rho)",RHO
390 INPUT "Angle to vertical line",THETA
400 X=FNPRX(RHO,THETA):GOSUB 130
410 PRINT "The horizontal length (X) is";X:GOSUB 150:GOTO 2000
420 ?"Polar to rectangular conversion, find Y":GOSUB 130
430 INPUT "Diagonal length (rho)",RHO
440 INPUT "Angle to vertical line (theta)",THETA
450 X=FNPRY(RHO,THETA):GOSUB 130
460 PRINT "The vertical length (Y) is";X:GOSUB 150:GOTO 2000
470 ?"Rectangular to polar conversion, find hypotenuse (rho)":GOSUB 130
480 INPUT "Horizontal length (X)",X
490 INPUT "Vertical length (Y)",Y
500 XX=FNRPR(X,Y):GOSUB 130
510 PRINT "The hypotenuse (rho) is";XX:GOSUB 150:GOTO 2000
520 ?"Rectangular to polar conversion, find angle (theta)":GOSUB 130
530 INPUT "Horizontal length (X)",X
540 INPUT "Vertical length (Y)",Y
550 XX=FNRPT(X,Y):GOSUB 130
560 PRINT "The angle (theta) is";XX:GOSUB 150:GOTO 2000
570 ?"Find the angle opposite side X using 3 sides of a triangle":GOSUB 130
580 INPUT "Horizontal length (X)",A
590 INPUT "Vertical length (Y)",B
600 INPUT "Diagonal length (hypotenuse)",C
610 X=FNSSS(A,B,C):GOSUB 130
620 PRINT "The angle opposite X is";X:Y=90-X
630 PRINT "The angle opposite Y is";Y:GOSUB 150:GOTO 2000
640 ?"Find the X root of a number":GOSUB 130
650 INPUT "Enter any number",X
660 INPUT "Enter root number",R
670 XX=FNROOT(X,R):GOSUB 130
680 PRINT "The ";R;" root of ";X;" is";XX:GOSUB 150:GOTO 2000
1450 ?"Find the reciprocal of a number":GOSUB 130
1460 INPUT "Enter any number",A
1470 X=FNREC(A):GOSUB 130
1480 PRINT "The reciprocal of ";A;" is";X:GOSUB 150:GOTO 2000
1490 ?"Convert degrees to radians":GOSUB 130
1500 INPUT "Enter number of degrees",A
1510 X=FNDEG(A):GOSUB 130
1520 PRINT A;" degrees equal ";X;" radians":GOSUB 150:GOTO 2000
1530 ?"Convert radians to degrees":GOSUB 130
1540 INPUT "Enter number of radians",A
1550 X=FNRAD(A):GOSUB 130
1560 ?A;" radians equal ";X;" degrees":GOSUB 150:GOTO 2000
1570 ?"Find the volume of a cube":GOSUB 130
1580 INPUT "Enter length of one side",A
1590 X=FNVOLC(A):GOSUB 130
1600 PRINT "The volume of the cube is ";X;" cubic measures":GOSUB 150:GOTO 2000
```

```
1610 ?"Find the volume of a rectangular box":GOSUB 130
1620 INPUT "Enter width of box",W
1630 INPUT "Enter length of box",L
1640 INPUT "Enter depth of box",H
1650 X=FNVOLR(W,L,H):GOSUB 130
1660 ?"The volume of the box is ";X;" cubic measures":GOSUB 150:GOTO 2000
1670 ?"Find the volume of a sphere":GOSUB 130
1680 INPUT "Enter the radius",R
1690 X=FNVOLS(R):GOSUB 130
1700 ?"The volume of the sphere is ";X;" cubic measures":GOSUB 150:GOTO 2000
1710 ?"Find the volume of a pyramid":GOSUB 130
1720 INPUT "Enter base area in square measures",B
1730 INPUT "Enter height of the pyramid",H
1740 X=FNVOLP(B,H):GOSUB 130
1750 ?"The volume of the pyramid is ";X;" cubic measures":GOSUB 150:GOTO 2000
1760 ?"Find the volume of a cylinder":GOSUB 130
1770 INPUT "Enter radius",R
1780 INPUT "Enter length of the cylinder",H
1790 X=FNVOLL(R,H):GOSUB 130
1800 ?"The volume of the cylinder is ";X;" cubic measures":GOSUB 150:GOTO 2000
1810 ?"Find the surface area of a cube":GOSUB 130
1820 INPUT "Enter length of one side",S
1830 X=FNSURC(S):GOSUB 130
1840 ?"The surface area of the cube is ";X;" square measures":GOSUB 150:GOTO 2000
1850 ?"Find the surface area of a rectangular box":GOSUB 130
1860 INPUT "Enter the width of the box",W
1870 INPUT "Enter the length of the box",L
1880 INPUT "Enter the depth of the box",H
1890 X=FNSURR(W,L,H):GOSUB 130
1900 ?"The surface area of the box is ";X;" square measures":GOSUB 150:GOTO 2000
1910 ?"Find the surface area of a sphere":GOSUB 130
1920 INPUT "Enter the radius",R
1930 X=FNSURS(R):GOSUB 130
1940 ?"The surface area of the sphere is ";X;" square measures":GOSUB 150:GOTO 2000
1950 ?"Find the surface area of a cylinder":GOSUB 130
1960 INPUT "Enter the radius",R
1970 INPUT "Enter the length of the cylinder",H
1980 X=FNSURL(R,H):GOSUB 130
1990 ?"The surface area of the cylinder is ";X;" square measures":GOSUB 150:GOTO 2000
2000 GOSUB 140:?"Menu:":GOSUB 130
2010 ?1,"Lesser of two numbers"
2020 ?2,"Greater of two numbers"
2030 ?3,"Average of two numbers"
2040 ?4,"Limit number of decimals"
2050 ?5,"Polar to rectangular, find horizontal length"
2060 ?6,"Polar to rectangular, find vertical length"
2070 ?7,"Rectangular to polar, find diagonal length"
2080 ?8,"Rectangular to polar, find angle"
2085 ?9,"Angles opposite two sides"
2090 ?10,"Root of a number"
2100 ?11,"Reciprocal numbers"
2110 ?12,"Convert degrees to radians":GOSUB 130
2111 ?"To choose one of the above, press >RETURN< "
2112 INPUT "To see other choices, press > Y < ",Z$
2113 IF Z$="Y" THEN 2120 ELSE GOSUB 130:GOTO 2230
2120 GOSUB 140:?13,"Convert radians to degrees"
2130 ?14,"Volume of a cube"
2140 ?15,"Volume of a rectangular box"
2150 ?16,"Volume of a sphere"
2160 ?17,"Volume of a pyramid"
2170 ?18,"Volume of a cylinder"
2180 ?19,"Surface area of a cube"
2190 ?20,"Surface area of a rectangular box"
2200 ?21,"Surface area of a sphere"
2210 ?22,"Surface area of a cylinder":GOSUB 130
2220 ?23,"Exit program":GOSUB 130
2230 INPUT "Which?",WHICH:GOSUB 140
2240 ON WHICH GOTO 200,240,280,320,370,420,470,520,570,640,1450,1490,1530,1570,
     1610,1670,1710,1760,1810,1850,1910,1950,160
```

Spread Sheets & Data Communications

by John Steiner

This month we will take a look at a newly released spread sheet program and also look at data communications for the Color Computer. Both of these applications are very popular among CoCo users, and they probably constitute most of my time on the computer.

Spread Sheets

One of the most popular classes of software for the microcomputer is the spread sheet. These useful programs have sold more microcomputers than any other type of software because of their versatility and usefulness. The CoCo user was not to be left out.

The first program available was Spectaculator by Radio Shack, which has many of the features of a spread sheet, but is missing quite a few of the more useful ones. C.C. Calc came along, and the first true CoCo spread sheet was available. The program, written in BASIC, has a relatively small sheet size and slow calculation speed, but many of the unique spread sheet features are there. Elite Calc was introduced at the April '83 Rainbowfest, and became the CoCo users first full fledged machine language spread sheet program. Though Elite Calc has its flaws, until recently it has been the only full sized spread sheet available for standard CoCo's. (Flex users have Dynacalc available. Though I have never seen it operate, the ads convey that it is indeed a full power spread sheet program.) *[Editor's Note: We use Dynacalc all of the time at MICRO for a very wide range of functions. It works very well.]*

VIP Calc

Softlaw, Inc. (Formerly Nelson Software Systems) has finally released VIP Calc. It's been a long wait (I remember asking for it at their booth at the April '83 Rainbowfest), but it was worth it. The powerful program is modeled after the granddaddy of all spread sheet programs, Visicalc. CoCo users who also have access to Visicalc on other systems will have no trouble becoming accustomed to VIP Calc.

There are improvements upon the original Visicalc, including sorting and setting individual column widths. Minor modifications have been made to take advantage of the particular hardware and keyboard features of the CoCo, however, so there are some differences in the command structure from the original Visicalc. Up to 16 display windows can be set to compare information. A bank switching technique allows up to 33K of spread sheet in a 64K computer. Up to fifteen digit precision can be selected, and trig functions have been included. One of the most useful features is a LOCATE command that can search a sheet for a specified formula or text entry.

A unique marketing strategy by Softlaw has eliminated the tape to disk version upgrade problem. VIP Calc, and many of their other programs include both tape and disk versions of the software. The "Combo" packaging is a nice feature.

VIP Calc's screen display is high resolution and the user can select between 32, 51, 64 or 85 characters per line. This makes for the largest Calc screen display for the CoCo yet. The nice display comes at a price, though. Choosing a high resolution display causes the loss of about eight thousand bytes of available spread sheet memory. In addition, it slows down the program because of the length of time it takes to write the screen display. An 8K display will take a lot longer to write than the standard 512 byte CoCo screen.

In fact, speed (or lack of it) is probably VIP Calc's major deficiency. After using Elite Calc, which is very fast in calculation and display, VIP Calc seems to move like a turtle. It is helpful to turn off the automatic calculation mode when you are doing data entry, then use the ! command to recalculate after the data are entered. If you choose the 32 character screen display, you will find that screen display update is much faster, as well. However, the speed problem is a relative thing and, if I had not run other spread sheet programs, I probably never would have made any comments about it. You get used to it.

One of the slowest spread sheet programs I have ever seen is MicroPro's Calc Star, which can take upwards of 45 seconds to recalculate a medium size sheet. I use Calc Star for much of my business work, however, and find that the recalculation time is not of any real significance. The other advantages of using Calc Star outweigh the speed problem. VIP Calc is nowhere near that slow in calculating and displaying data, and it is a lot more powerful. I may end up changing to VIP for my company work.

One other disadvantage of VIP Calc for 32K only users is that some features were left out to conserve memory. Locate, Edit and the high resolution graphics screens are not available to 32K systems. This should give you enough incentive to make the jump to modify your system to 64K. VIP Calc will not run in a 16K computer.

I really enjoy using a spread sheet program for creation of numeric and even text data files. So many things can be done easily on a spread sheet that would take hours of programming time if you were to try to write a BASIC program to do the same thing. If you haven't had the opportunity to look into what a spread sheet can do for you, check it out. You might find that it can be a help in your daily work. For more details, and a simple program to introduce you to spread sheets, check out issue number 67

of Micro, December 1983. That issue was devoted to the spread sheet, and includes "MicroCalc", a spread sheet program for the CoCo.

CoCo Communications

The Color Computer makes a great Videotex terminal, as many people have already found out. Terminal software is inexpensive, and Modems are becoming much more reasonable in price. Two useful, yet inexpensive Modems that work well with the CoCo are the Mura MM-100, and the Anchor Automation Volksmodem. The Mura retails for $99.95, while the Volksmodem retails for $79.95. I have seen them both advertised for less in mail order ads.

The biggest problem is in configuring a cable that works with them. The CoCo has only four of the 25 standard RS-232 lines. The Mura modem has a 25 pin standard connector, while the Volksmodem has a five pin DIN connector.

When connecting these, or any modem for that matter, there are really only four required lines for an RS-232 port. They are TXD (transmit data), RCD (received data), GND (ground), and CD (carrier detect). The CoCo RS-232 port contains all these lines and, to work with any modem, they must be connected properly to the same lines on the modem connector.

One concept that has caused confusion in the past is the connection of RCD and TXD. Many people would make the assumption that RCD on the modem should be connected to RCD on the computer, and TXD would be connected likewise.

The connections won't work that way, however, since modems are usually wired as data sets, and computers are wired as data terminals. The difference this causes makes sense, though, so connection is made by putting the RCD line on the modem to the TXD line on the terminal. Similarly, the TXD line on the modem goes to the RCD line on the terminal. All other lines connect directly from the modem to the terminal.

Computer communications is an interesting aspect of the microcomputing hobby, and is becoming more and more useful in the world of business. The Dakota Database Bulletin Board System that I have been running since July of 1983 is still going strong, and over 3700 calls have been made to the system since it has been on line.

Last month we added an upgraded software package, and the BBS is more sophisticated in its message handling, uploading and downloading of programs and files, and general system operation. If you have a modem package, give the Dakota Database a call at 701-281-0233. It is online 24 hours a day, except for occasional periods of updating, and contains several Color Computer programs that you may download at no charge. I'll be looking for your message on the BBS.

And, farewell. MICRO has decided to discontinue microcomputer specific columns in favor of topic specific columns, so this is the last time **CoCo Bits** *will appear. I wish to thank all of you who have responded to this column for your comments, advice and general support. You may see me again soon in MICRO with a column on Telecommunications.*

MICRO

compass

by Loren Wright

Report on TPUG Conference

A lot happened at the 3rd Annual Toronto PET Users Group Conference. Brad Templeton, author of the utility package POWER and the assembler PAL, demonstrated a program development system that won't let the user make a programming mistake. For instance, in Pascal if you decide to write a PROCEDURE, the system will automatically provide the ENDPROC statement and prompt you for variable declarations, parameter lists, and such. Jim Strasma, Editor of The Midnight Gazette and Contributing Editor for MICRO, spoke on what to look for, and what to avoid, in commercial software. He emphasized selecting where you buy a product and evaluating product warranties.

Featured at the Saturday night banquet were VIC-based "Randy" robots, "Uses for a Dead Computer" by Transactor Editor Karl Hildon, and reminisces from TPUG founders Jim Butterfield and Lyman Duggan.

What About the 264?

Jim Butterfield and Jim Strasma had a lively discussion on the merits of Commodore's newly announced 264 Computer. As if on cue, someone showed up with a 264 — straight from Commodore Canada. Butterfield soon had it hooked up and running. Surprisingly, the 264 is even smaller than the C-64. It has arrow-shaped cursor control keys arranged in a diamond pattern. There are three differences obvious just from the power-up message. First, there are over 60K bytes available for BASIC. Second, the BASIC is version 3.5. Third, a message appeared that said: "SUPERSCRIPT on key 1". When Jim pressed function key 1, the word processor appeared instantly — a demonstration of the 264's built-in software capability. Another function key read and displayed the disk directory. The eight function keys can each be assigned with a simple statement.

The 264 uses a processor called the 1701. Contrary to what I said in a previous column, this is a member of the 6502 family and uses the same mnemonics. It still has only a 64K address space, but due to a more sophisticated architecture, is able to keep most of its operating system "floating."

The BASIC is more powerful, including convenient disk commands, graphics and sound statements (instead of POKEs), and structured programming statements. There is also a built-in, extended, machine-language monitor, and — first time for Commodore since the KIM — a reset button!

On the minus side, there is no numeric keypad, cassette capability is gone, the sound is primitive (compared to the C-64), and there are no sprites. There should be 80 columns for a "productivity" machine. There are 16 different colors in 8 different luminances.

Commodore's emphasis in marketing the 264 will be "productivity," and, it's true, this machine is a lot easier to operate. My work would definitely go faster. Having a structuring capability available on power-up is a real treat. Also, the built-in extended monitor and reset button can relieve a lot of frustration in assembly language program development.

Despite all the productivity-oriented features, the actual hardware is a little disappointing. This machine seems to be aimed somewhere in between the IBM/IBM Clone market and the low-end market. Butterfield, going along with the productivity emphasis, thinks there's a place for the 264, but Strasma thinks it may be a case of "too little, too late". Due to its limitations in graphics and sound, there may not be much recreational software for it. I suspect the C-64 will still be alive and well a year from now. It appears that the 264 will really be introduced, but certainly not while the C-64 is doing so well. It doesn't really compare with the C-64, but will people be able to tell? The pricing is still uncertain ($500?), and there may be some changes before it comes out. A numeric keypad and 80-column word processing would sure help productivity!

COMAL for the Commodore 64

I recently purchased a COMAL disk from the COMAL Users Group and did a little playing with the graphics commands. I had reviewed a previous version of COMAL for the PET and saw some promise there. It is a structured language, yet without the picky syntax requirements of Pascal. It has most BASIC commands, as well as REPEAT...UNTIL, WHILE...ENDWHILE, FOR...NEXT, IF...THEN...ELSE...ELIF...ENDIF, and a good CASE structure. Like BASIC, it is easy to program because you don't have to deal with an editor, compiler, and P-code interpreter to get your program to run. All you have to do is edit the program and RUN it. This is particularly important in a learning situation. There were problems with the PET version I saw, such as a "split" interpreter, and a few minor bugs. Also, since the PET had its ROMs hard-wired in, COMAL had to be loaded on top of BASIC, taking up most of the valuable RAM. When I began seeing ads for the C-64 version, my interest was rekindled. I knew the architecture of the C-64 would be better suited to a COMAL implementation.

COMAL has been around for quite a while, but it has received little attention in the U.S. In Europe it has a strong following. It is the official teaching language in Denmark, Ireland, Sweden, and Norway, and will be soon in other countries. It was designed by Borge Christensen as a combination of the "best of Pascal and BASIC." The C-64 version boasts a built-in turtle graphics system, so its promoters now add Logo to their "best of" list.

COMAL works much the same as BASIC, in that you can execute commands in the immediate mode, and you can test each procedure or function as you write it. However, it actually is a little more complicated. The system makes three passes through a program. Syntax errors are detected as you enter each line. Then, when you RUN the program, two more passes are made. It sounds complicated and time consuming, but overall COMAL averages out to about three times faster than BASIC. For string manipulations, COMAL is much faster than that. The string functions are simpler, but more powerful, than those of Microsoft BASIC. For instance, you can assign a substring — not allowed in BASIC.

The graphics commands are very convenient and they work on either hi-res or multicolor bit-map screens. There is very little required to set up a bit map screen since the system is designed to operate in bank 3, the $C000 block and the RAM under the I/O area and Kernal ROMs. All the memory manipulation is taken care of automatically. There's also a whole set of sprite commands, but these are designed to work with a bit map screen. I was able to get a few sprites to work in bank 0 with ROM characters. There doesn't seem to be any easy way to use programmed characters, although it is possible. Part of the problem is that there isn't any published memory map or other documentation of COMAL's inner workings, although these are in the works. Another problem is that there is less memory available for machine language programs. One promising feature is that position-independent routines can be programmed as strings, a technique popular with advanced Atari programmers.

At the TPUG Conference, I met Len Lindsay, who runs COMAL Users Group, U.S.A., Ltd. (5501 Groveland Terrace, Madison, WI 53716). The users group owns the copyright to the disk version and distributes copies for $19.95. He does allow individuals to freely copy the system disk, as long as they don't make any changes and don't make any money off it. Even large users groups may distribute copies, but they must check first with Len. Len is the major proponent of COMAL in the US, and he has been with it since nearly the beginning. He was founder of *The PET Gazette* (the most successful of several newsletters and magazines that were combined to form **COMPUTE!**), and is working to get COMAL established as a major force in the U.S. With a little help from enthusiastic users and a computer company or two, COMAL will take over even faster.

Support for COMAL is growing. The COMAL Users Group publishes a regular newletter called *COMAL Today* ($15/year), and an applications tutorial book series *Captain COMAL* ($19.95 with disk for each book). In addition, there is a reference book, the *COMAL Handbook* by Len Lindsay ($18.95), and a number of books from European publishers. All of these, including the European books, are available from the COMAL Users Group, and many are available in computer and book stores.

A Better COMAL — How long will Commodore sit on it?

There is a better COMAL, but it is not available. Commodore owns it, and is taking its time with producing it. The disk-based COMAL reviewed above only leaves you with 10K free for your program (equivalent to 16K in BASIC terms). The new, cartridge-based COMAL will leave over 30K free, as well as add many new features and commands. One feature I have heard described is a multi-level error-trapping system. Although I don't know for sure, there should be a little more flexibility added to graphics programming. Len Lindsay, and a few others, have seen this new version, and are very impressed.

Like other companies, Commodore has limited manufacturing facilities, and the ROM burners required to make cartridges are particularly in demand for things like hot-selling games. For Commodore, it's a matter of juggling priorities and demands. Look for the COMAL cartridge in late summer or fall. In the meantime, you can buy the disk version, and learn all about the language. It will take a while to exhaust the 10K limit. Then keep up the pressure on dealers, so they can keep pressure on Commodore. It might work!

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 60, CHELMSFORD, MA

POSTAGE WILL BE PAID BY ADDRESSEE

MICRO™

P.O. Box 6502
Chelmsford, MA 01824

SAVE 20%!!!!

Subscribe to MICRO

Use This Postage Paid Card to Order the Next 12 Issues of MICRO and SAVI

$6.00 Off Newsstand Price!

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 60, CHELMSFORD, MA

POSTAGE WILL BE PAID BY ADDRESSEE

MICRO™

P.O. Box 6502
Chelmsford, MA 01824

A Feast O Computin Ideas...

New Book From MICRO

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 60, CHELMSFORD, MA

POSTAGE WILL BE PAID BY ADDRESSEE

MICRO™

P.O. Box 6502
Chelmsford, MA 01824

Order These Best Sellers From MICRO

A Major Hardware Interface

by Ralph Tenney

Figure 1

1	STROBE* (DATA READY)
2	DATA1
3	DATA2
4	DATA3
5	DATA4
6	DATA5
7	DATA6
8	DATA7
9	DATA8
10	ACKNLG* (PRINTER READY)
11	BUSY
33	PRINTER GROUND
36	SLCT IN* (TIE TO GROUND TO SELECT PRINTER)

Figure 1. Pinout of Centronix-style parallel input plug.

Last time we began looking at the problem of connecting my 32K CoCo as a printer buffer for the 64K CoCo and my Commodore 64. A sketchy outline of the planned system was shown, and a simple RS-232 interface between a CoCo and the C-64 was shown. This month, let's do some detail designing of the major hardware interface — the receiver board for the 32K CoCo.

The most often overlooked item in hardware or software is a complete design specification. If you omit the spec while working for another person or company, two problems can arise. First, you may overlook a feature you verbally agreed to, and need a expensive re-design (at your expense). Second, the customer can say, "But I told you I wanted another printer port," or something similar. With an agreed-to and signed design specification, you have the opportunity to charge extra for changes made after the spec is signed off. In working for yourself, (always after a long day at work), it's easy to overlook something. If you have your own spec to work against, you can trade off bells and whistles against your own time and pocketbook if you decide to make a change. The point is that the project is under some control and won't "grow like Topsy" unless you decide to let it. Besides the other advantages, the design spec is an good start on the documentation you *must* have!

Multi-Port COCO Interface

I. Minimum Interface Capability

A. Centronics Compatible Parallel Ports

1. Port #1 must have a female Centronics style connector with pinout conforming to that of Figure 1. This port is an input and must accept a negative-going strobe pulse .5 uSec. wide (minimum) as a "data ready" signal and return a similar "acknowledge" pulse when ready to accept another data byte. An active-high "Busy" signal shall be provided in addition to the "data ready" and "acknowledge" signals. This port may be implemented as a cable terminating in the requisite connector.

2. Port #2 shall be output only, using a connector which can mate with the connector of Port #1. The pinout shall conform to that of Figure 1. This port must issue a negative-going "data ready" signal .5 uSec. (minimum) wide and recognize a similar "acknowledge" pulse. The port shall also monitor a "printer busy" input and hold off further output until "busy" has gone inactive. This port must be implemented with cable of 15" minimum length.

B. RS-232 Serial Ports

1. Port #3 shall be an RS-232 input with selectable baud rates of 600, 1200 and 2400. An output line shall be provided for, with pinout to match the Radio Shack Color Computer serial port as implemented by the 1.1 BASIC ROM. A "busy" signal shall be returned on the fourth wire of the cable.

2. Port #4 shall be an RS-232 output only, with selectable baud rates of 600, 1200 and 2400. This channel shall have a unique output connector and be responsive to a "busy" input. The data normally assigned to the channel shall be capable of being diverted to the output of Port #3.

II. Physical and Electrical Considerations

A. The interface card shall be physically compatible with the cartridge expansion port of the Radio Shack Color Computer and shall have external support provided for the end of the card opposite the connector. Cables used as input and output adapters shall be removable by plugging onto single- or double-row header strips.

B. The ports shall be based in the CoCo expansion I/O area between $FF40 and $FF7F and the controlling software must auto-start and be based at $8000. A separate specification will define the functions of this software.

C. The two parallel ports shall be built with a 6522 VIA, plus a separate latch to generate the local "busy" signal and input logic to monitor the remote "busy" signal. A 6850 ACIA will be used for both ports with switch logic to divert the output channel as needed. An acceptable alternative is to create two full duplex channels with two ACIAs.

Implementation of the Interface

Let's study the CoCo expansion port to see what we have to work with. Table 1 gives the pinout of this 40 pin connector. The application of most of the signals is obvious, but there are a few signals unique to the 6809/6883 architecture of the Color Computer. "E" is the

primary clock signal for the 6809, and Q is similar to the Phase 2 6502 clock, except that it is a quadrature clock which gives extended timing margins for interfacing. CTS* is the main SELECT signal for the cartridge port, and is active for the the range $C000 to $FF00. SCS* is a secondary SELECT strobe active between $FF40 and $FFCO and is intended to select I/O devices. SLENB* is furnished to entirely disable the internal decoding, which allows plug-ins such as the Z-80 module which runs C/PM software. HALT* stops the 6809 and CART* is the line which allows game cartridges to auto-start. (A second auto-start mechanism is available under Extended BASIC — if the bytes at $C000 and $C001 are "DK", control is handed over to the cartridge.) It is important to note that none of the processor address or data lines are buffered. The E and Q lines are generated by the bipolar 6883 and the two SELECT lines are generated by a low power Schottky IC. So, after choosing which IC is needed in the interface, we will need to examine the bus loading to be sure that the interface will not overload the computer.

TABLE 1 — CoCo Expansion Port

Pin	Function	Pin	Function
1	-12 Volts	21	Address A2
2	+12 Volts	22	Address A3
3	HALT*	23	Address A4
4	NMI*	24	Address A5
5	RESET*	25	Address A6
6	E (clock)	26	Address A7
7	Q (clock)	27	Address A8
8	CART	28	Address A9
9	+5 Volts	29	Address A10
10	Data D0	30	Address A11
11	Data D1	31	Address A12
12	Data D2	32	CTS* (SELECT 1)
13	Data D3	33	Ground
14	Data D4	34	Ground
15	Data D5	35	Sound input
16	Data D6	36	SCS* (SELECT 2)
17	Data D7	37	Address A13
18	R/W*	38	Address A14
19	Address A0	39	Address A14
20	Address A1	40	SLENB*

The interface board needs the following capability in addition to the specified I/O ports:

1. A clock source to generate the clock frequencies needed for the three specified baud rates.
2. Decoding for the VIA, one or two ACIAs and at least one latch.

The ACIA (Asynchronous Communications Interface Adapter) uses one of two clock frequencies: 16 x the baud rate (/16 mode) and 64 x the baud rate (/64 mode). If you accept the software overhead required to change the ACIA divide ratio, the ACIA can be operated in both /16 and /64 modes with two input frequencies and gain one additional baud rate:

38400 Hz./16 = 2400 baud
38400 Hz./64 = 600 baud
19200 Hz./16 = 1200 baud
19200 Hz./64 = 300 baud

Figure 2 shows how these clock frequencies will be generated. The CD4024 is a 7-bit binary counter, and the CD4068 is an 8-input AND/NAND gate used to decode the counter output lines. The clock input to the eounter is

Figure 2

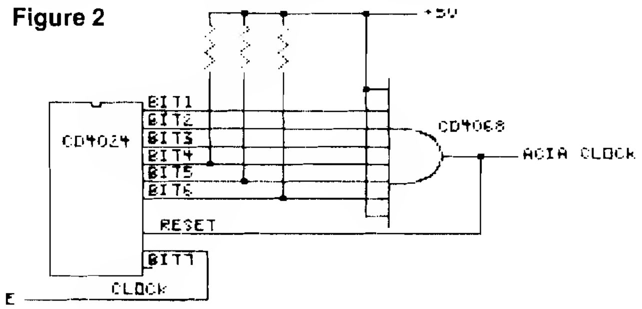


Figure 2. Counter and switched decoder circuit generates baud rate clock signals.

the 6809 "E" clock whieh runs at 895 kHz. Table 2 shows which counter outputs need to be decoded to generate the two frequencies. Note that three outputs are common between the two decoding schemes, so three counter outputs need to be selected or deselected to program the counter.

TABLE 2 — Counter Coding

895 kHz/23 = approx. 38400 Hz
895 kHz/46 = approx. 19200 Hz

23 = $17 = 00010 111
46 = $2F = 00101 111

Decode bits 0, 1 and 2 are common;
Bits 3, 4 and 5 must be switched to select frequency.

ACIA and VIA Decoding

Table 3 shows the decoding requirements for the 6522 VIA and the 6850 ACIA. The ACIA occupies only two memory locations and the VIA occupies 16 memory locations. You read the chart this way: CS (Chip Select) lines are shown true for a seleeted chip; any change will deseleet the chip. An active-low CS line is typically driven by the decoder, while others are permanently selected or driven by a higher-order address line. RS lines work together to select individual registers, and are usually driven by low-order address lines. The ACIA uses the RS and R/W* lines together to cram four registers into two address bytes. The price is that each register is read-only or write-only, which complicates the programming as will be discussed later.

TABLE 3 — I/O Address Decoding

6850 ACIA

Register	RS	R-/W*	CS0	CS1	CS2*
Control Reg	0	0	1	1	0
Status Reg	0	1	1	1	0
Transmit Reg	1	0	1	1	0
Receive Reg	1	1	1	1	0

6522 VIA

Register	CS1	CS2*	RS3	RS2	RS1	RS0
I/O Reg B	1	0	0	0	0	0
I/O Reg A	1	0	0	0	0	1
DD Reg B	1	0	0	0	1	0
DD Reg A	1	0	0	0	1	1
Counter 1 Lo	1	0	0	1	0	0
Counter 1 Hi	1	0	0	1	0	1
Ctrl Latch Lo 1	0	0	1	1	0	
Ctrl Latch H1 1	0	0	1	1	1	
Counter 2 Lo	1	0	1	0	0	0
Counter Hi	1	0	1	0	0	1
Shift Reg	1	0	1	0	1	0
Aux Ctl Reg	1	0	1	0	1	1
Per Ctl Reg	1	0	1	1	0	0
Int Flg Reg	1	0	1	1	0	1
Int Ena Reg	1	1	1	1	0	
I/O A/NH	1	0	1	1	1	1S1

Figure 3 shows a 74LS138 decoder with the required conneetions to create memory "slots" for a number of I/O devices, and defines the memory locations for the VIA, ACIAs and a latch to be used on the interfaee. The 74LS138 is a three-line to eight-line decoder whieh has one active-high SELECT line (G1) and two aetive-low SELECT lines (G2A and G2B). If all the SELECT lines are true, the output addressed by the binary values on the three input lines (A, B and C) is active-low; all others are inactive. If any of the SELECT lines are false, all outputs are inactive. The primary SELECT is SCS*, which must be low (G2A). By requiring Bit 7 to be low (G2B), the decoder is inactive above $FF7F. With Bit 6 driving G1 and C, only the outputs 4, 5, 6 and 7 will be active. The selected devices and the effective addresses are shown in Figure 3. Note that the VIA entirely fills its assigned slot, but that the other devices will respond multiple times in their own slots.

Figure 3

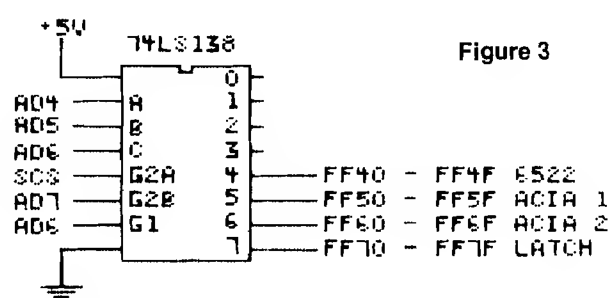


Figure 3. 74LS138 decoder circuit with resulting memory blocks and assigned function.

Title: Overcoming Computer Illiteracy
Authors: Susan Curran and Ray Curnow
Price: $12.95
Publisher: Viking Penguin Inc.

Written for those who have no previous knowledge of computers, this is Penguin's first foray into computer-related books. It is divided into five sections: Aspects of Modern Computer Technology, Manipulating Symbols, Development of the Computer — An Historical Perspective, Programming and Languages, Applications of Larger-Scale Computer Systems. The style of the writing is narrative, with no requirement of previous knowledge of computers, mathematics, or electronics. Simple programs are provided that can be used by beginners. The basic principles of data storage and manipulation are explained. The 6502 chip is chosen as an example for discussing the basic components and workings of microprocessor chips. Languages examined include machine language, BASIC, FORTRAN, COBOL and other high-level languages. Computer aided design, artificial intelligence, meteorological uses are among the applications discussed. A glossary and bibliography are provided. Points are illustrated through drawings, charts and tables.

Level: Beginner.

Title: Introducing the Acorn/BBC Micro
Author: Ian Sinclair
Price: $12.95
Publisher: Prentice-Hall, Inc.

Mr. Sinclair draws upon his many years of experience as a teacher and author in writing this introduction to the Acorn. As stated by the author, the Acorn was designed to a very advanced specification. Bearing this in mind he set to teach a beginner on a machine not well suited to beginners. The book is aimed at this level and makes no pretenses otherwise. Starting with instructions on how to set up your new Acorn, everything from tuning your TV to dealing with cables is dealt with. The programming concentrates on BASIC, neglecting those commands that might be common to other micros but are not part of the Acorn. The fundamentals of sound, graphics and color are shown, omitting those details that are too technical to be of use to a beginner. This approach of avoiding overly technical and lengthly discussions carries throughout the book. The examples are short and concise, saving the beginner from unnecessary confusion. The appendices cover Reserved Words, Cassette Capers, Appending Programs, Magazines and User Groups, and the Cassette Bug Fix. The latter is aimed at correcting a bug present in the 0.1 system's cassette. The book and programs are useful for both the 0.1 and the newer 1.2 operating system.

Level: Beginner.

Title: 8-Bit & 16-Bit Microprocesser Cookbook
Author: Joseph J. Carr
Price: $13.50
Publisher: Tab Books, Inc.

A reference and sourcebook that guides the reader in the technical realm of microprocessors. Offering much valuable information that is not easily found elsewhere, a variety of topics are covered: architecture, individual chip characteristics, handling of interrupts, timing, control signals, interfacing memory, pinouts, variations, interfacing I/O and the instruction sets. Of the different 8 and 16-bit microprocessors examined, the Z80 and 6502 are given particular attention and detail. Other 8-bit chips covered are Motorola's 6809 and 6800 and Intel's 8080A, 8085A, 8086, and 8088. The two 16-bit microprocessors that are given the most attention are the MC68000, Motorola's bid for the 16-bit market, and Intel's iAPX86/10. The appendices look at address decoder circuits and techniques, generating device-select pulses, input/output devices, and low-voltage DC power supplies. Certainly this field is too vast for this book to be comprehensive, but it definitely provides the reader with enough information to become knowledgable about these more popular microprocessors.

Level: Intermediate to advanced.

Title: The Anatomy of the Commodore
Authors: Michael Angerhausen, Dr. Achim Becker, Lothar Englisch and Klaus Gerits
Price: $19.95
Publisher: Abacus Software

Starting with machine language programming, this look at the C64 as an all-around computer is a valuable guide. It begins with the Monitor and its uses, naturally progressing into the in's and out's of machine language programming. Then assembler programming is discussed, with a table of 6510 commands. Next, a more in-depth look is taken into memory configurations, the expansion and user ports and other special features of the 6510 microprocessor. Sound and graphics programming are covered, examining the Sound Controller 6581 and the Video Interface Chip 6567. The Analog/Digital Converter is explained, along with how to handle it. Sprite graphics are discussed in some detail, including capabilities, structure, and programming. The BASIC interpreter is viewed from a number of vantage points. A comparison table of Vic-20, C64 and CBM/PET ROM addresses is provided. The last chapter deals with input/output control — CIA 6526. The appendices consist of a ROM listing, a short lesson in Hexadecimal arithmetic, a summary of capabilities and a bibliography.

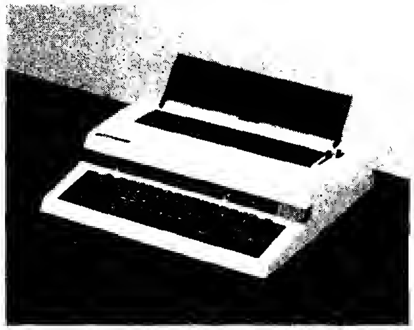
Level: Advanced beginner to Advanced.

Name: **Enstat Printer Mat**

Description: Dissipates static and absorbs sound and vibration from printers, electronic typewriters, etc. This single layer mat's positive static drain to ground functions in all temperatures and humidity conditions found in office or home environment and works with people and other conductive objects as well as machines. Sized to fit most tabletop equipment (18" x 22"), it has a 10 ft. ground cord with one megohm resistor.

A "bottom feed" design allows printer paper to pass over the mat edge, discharging static before it enters the printer.

Price: $44.95
Available: Semtronics
P.O. Box 599
Scotch Plains, NJ 07076
201/561-9520



Name: **McMill**
System: Apple II, II+, IIe

Description: This 68000 coprocessor card is an excellent entry level educational board for those interested in working the 68000. It uses Motorola's 60008 processor which is totally code compatible with 68000.

Included are complete hardware documentation, schemata, and a FIG FORTH software. Optional software includes a 68000 cross assembler from SC Software which includes efficient debugging of code with simple trap monitor and built-in line oriented editor, and an enhanced screen editor for faster programming. McMill comes with a one-year hardware warranty.

Price: $229 ($299 with Assembler)
Available: Stellation Two
P.O. Box 2342
Santa Barbara, CA 93120
805/966-1140

Name: **Flexible Head Cleaning Disk**
Hardware: 8", 5 1/4", or 3 1/2" drive

Description: A floppy disk head cleaner that dry cleans without abrasives and requires no liquid solution which might leave residue. Removes Ferric Oxide contamination and traps debris internally in special pockets. Just insert into drive and run for 30-60 seconds weekly. Each Disk provides 30 cleanings at about $.83 per cleaning. Available in three sizes.

Price: $24.95 (5 1/4" disk)
Available: Vikor Company
P.O. Box 3123
Nashua, NH 03061
603/889-8530

Name: **PaperClip**
System: Commodore 64

Description: This word processor allows a switch to 80-column display with no additional hardware, just a simple common routine. Horizontal scrolling up to 250 columns across for wider documents or tables also is possible. A powerful search capability uses unusually precise definition of "search" words. A unique printer set-up routine allows the user to get the best advantage of any printer, and support files include over 100 popular printers.

PaperClip also can be used with SpellPack, a spelling checker program with an internal dictionary of over 20,000 words.

Price: $89.95
Available: Batteries Included
3303 Harbor Blvd, Ste C9
Costa Mesa, CA 92626
714/979-0920

Name: **Bank President**
System: Apple II, IIe, IIc, Macintosh, IBM

Description: This first title in a series is designed to teach the fundamentals of business strategy and decision making through role-playing games. As CEO, the player formulates strategy and makes decisions that determine how well the company performs. Users can play the game alone, against the computer, or in competition with other players. Actions of one competitor can affect the performance of another, as in real life. Players are CEO of a large commercial bank of any type they want, setting loan and dedposit interest rates, raising or lowering employee salaries, investing capital and opening branches. Over 70 charts and graphs keep the player informed of the economy, bank conditions and competitors' actions. There are three levels of play.

High-Tech Entrepreneur and Venture Capitalist are the next two titles due in the series.

Price: $74.95
Available: Lewis Lee Corp.
P.O. Box 51831
Palo Alto, CA 94303
415/853-1220

Name: **Simulated Computer**
System: Atari, Commodore 64

Description: An award-winning simulation of the inner workings of a computer. The program takes you on a trip through an imaginary computer, revealing the secrets of machine and assembly language programming. You create a program and then see and hear the flow of data as it travels into memory, as registers are modified, and as the CPU processes information. The package has programmable sound and a graphics "turtle screen."

Price: $
Available: EduSoft
P.O. Box 2560
Berkeley, CA 94702
1-800-EDUSOFT

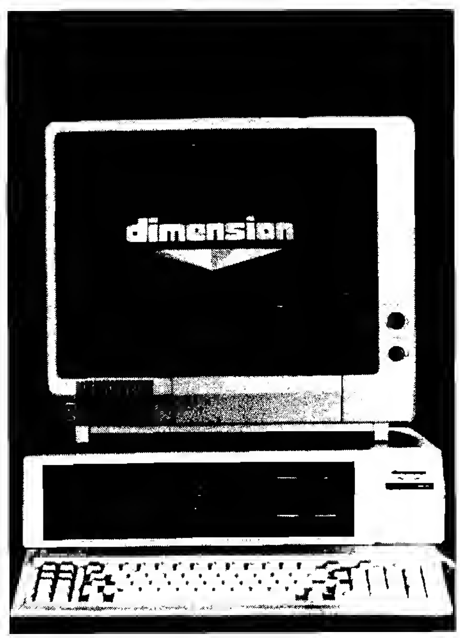
Name: **Flying Colors**
System: Commodore 64 (also Apple IIe/II+ versions)
Hardware: Joystick

Description: A color graphics software package designed for use with a standard joystick. A windowed screen menu lets the user pick the desired functions for drawing. Choices include thick and thin lines, automatic circles and boxes of any size, erasures, and ability to fill enclosed areas with a variety of colors. Drawing speed can be adjusted for exacting detail work and different colors and brush sizes are available for painting. Text can be added anywhere to the screen and a grid helps align the pictures. Pictures can be saved and retrieved from disk.

A sophisticated Slide Projector program is also included so users can create their own slide shows for business presentations and recreation.

Price: $39.95
Available: The Computer Colorworks
3030 Bridgeway
Sausalito, CA 94965
415/331-3022

Name: **Dimension 68000**
Memory: 256K RAM (512K optional)
Language: Virtually all



Description: This amazing machine comes with a display that handles up to 100 characters by 48 lines in 16 colors, and two 5 1/4" floppies, serial, parallel and game controller interfaces, and a six-slot internal expansion but that makes possible expansion to 12.5 megabytes of RAM and even more.

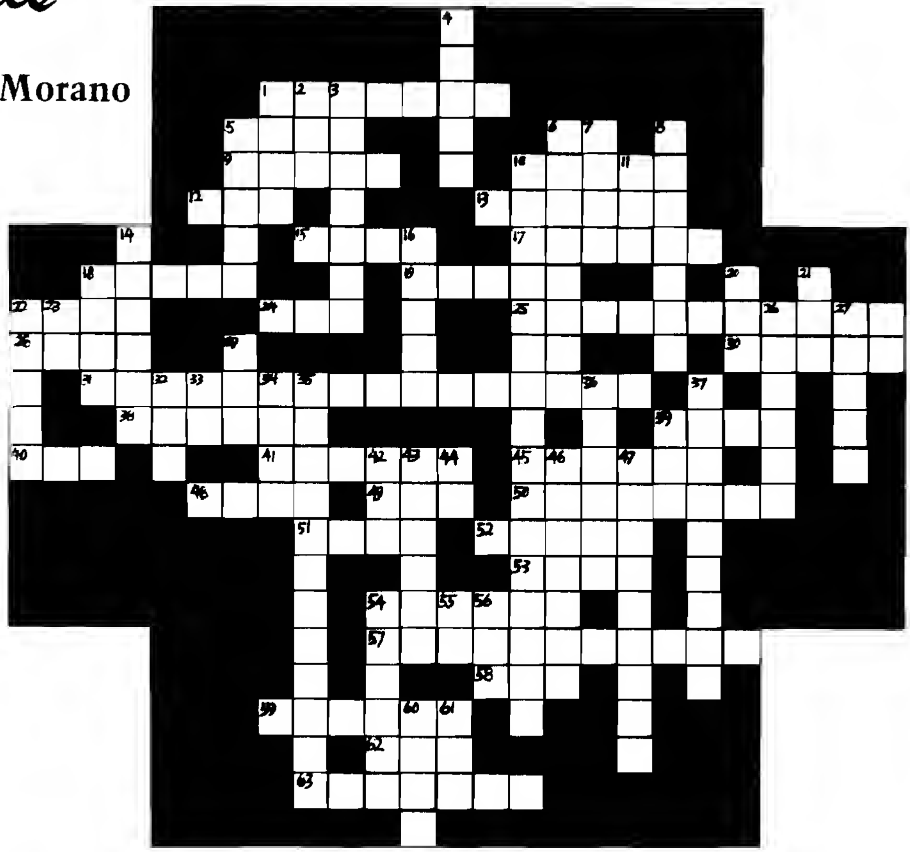
However, the most exciting part is that this computer can run almost any company or microprocessor's software. Plug-in CPU cards automatically reconfigure the disk drive format and video display to imitate the hardware of the simulated machine. CP/M-68K is standard, but p-System, Idris, UNIX are all possible, as are Apple, IBM, KayPro and other CP/M and MS-DOS software.

Price: Under $4000
Available: Micro Craft Corporation
4747 Irving Blvd.,
Suite 241
Dallas, TX 75247
214/630-2562

lyte bytes

by Mark S. Morano

C
R
COMPUTER
S
S
W
O
R
D
S



ACROSS

1. A hackers ham.
5. If not less, than ________.
6. Strength of light given off by candles (abbr).
9. To wipe out.
10. A printers favorite flower.
12. Branch on not equal (abbr)
13. ____________ Bowes.
15. In heaven they are pearly (singular).
17. A kind of graphics.
18. What tax evaders and programmers have in common.
19. Dot.
22. A ________ off the old block.
24. A feline.
25. The study of controlling an industrial process automatically by computer.
28. Composer with speech impediment, Franz ________.
30. Cold hard __________.
31. A dying impersonator
38. A kind of race that impulses play.
39. Don't touch the third one.
40. A shortened room.
41. A sign that sounds like a crash.
45. A chauffer.
48. If — ________ — else.
49. A TV by any other name.
50. You need to do this to vote.
51. Secret ________.
52. Photo color with brownish tint.
53. Transported with delight.
54. What the Apple programmer exclaimed when he found out his assembler was missing.
57. What Bob Hope's annual performance rate has dropped to.
58. Minus without the vowels.
59. A description of space games.
62. What a programmer puts on when his program bombs.
63. A bust builder.

DOWN

1. What is leftover after eating an apple.
2. What holds a woman in place.
3. To begin again.
4. One of the 3 R's.
5. An orderform for restaurants.
6. What an instigator is.
7. Needles and ________.
8. 'The books must be done _____ __________ end or else!'
10. What a mail clerk does best.
11. Ready, ______, go!
14. Our Lady of Grace.
16. A dance for those who change their mind.
18. What Santa checks twice.
20. A manufacturer that sounds like the floor of a ship.
21. What nervous programmers develop.
22. On a __________ day.
23. A greeting.
26. A suitor.
27. Both astrology and hit records have this in common.
29. A group of professional hide and sneakers.
32. Hi______ (abbr.).
33. Machine Language (abbr).
34. What crashes, rises and drops, (abbr).
35. A form of communication using signals to start and stop.
36. Where something starts.
37. ________________ coordinates.
39. Lo______.
42. A coding scheme that sounds like a brand of underwear.
43. A state where apples grow.
44. Light (abbr).
46. Over and over, again and again.
47. A way computers talk.
54. An unusual state of affairs.
55. Luxury Edition (abbr).
56. Snow White (abbr).
60. Internat'l Conference on Information Processing (abbr).
61. Conditional Transfer of Control (abbr).

Advertiser's Index

Solving Lyte Bytes

In the July issue, Lyte Bytes had a computer word puzzle which required unscrambling the words and then arranging letters to find the answer. Below is the proper solution.

BECAUSE/HE/LOST/HIS/DRIVE.

NCEMMUOIN	MNEUMONIC
LXEPI	PIXEL
VICRSEURE	RECURSIVE
RPHEOP	HOPPER
EDPCOUESOD	PSEUDOCODE
GRMIOTLAH	ALGORITHM
CIKDOLRG	GRIDLOCK
CAUPDETIL	DUPLICATE
ITUNENOMLITOEMCAC	TELECOMMUNICATION
RIBYAN	BINARY
ILECDHAMXIE	HEXIDECIMAL
CBYISOML	SYMBOLIC

MICRO Reader Survey

The Editorial Staff and Advertising Staff of MICRO want to sincerely thank the hundreds of readers who have taken the time and made the effort to complete and return the **READER SURVEY** that was included in the June issue. We are currently working to encode and analyze the information that you provided. Although it will be a couple of weeks before the results are fully 'digested', this information has already started to influence our editorial directions and has been helpful in defining our readership to the advertising community.

If you have not yet returned your questionnaire, please do so before the end of August so that your interests will be represented.

Coming in September —

As announced in previous editorials, MICRO is extending its coverage into some new areas, some of which are represented in the September issue by:

- ☐ An Introduction to FORTH,
  by Kenneth Butterfield
  The basic whys and wherefores of this language
- ☐ A Structure Tree Utility in FORTH,
  by Mike Dougherty
  Produces "road-maps" of FORTH applications
- ☐ Multi-Tasking in FORTH,
  by Kenneth Butterfield
  How to implement a Multi-Tasking system
- ☐ 68000 Exception Processing — Part 1,
  by Mike Rosing
  Taking care of software and hardware 'exceptions'

Our coverage of the 6502/6809 world continues with:

- ☐ Graphic Printer for C64 — Part 3,
  by Michael Keryan
  Add full color capabilities to your printer
- ☐ Time Series Forecasting,
  by Brian Flynn
  Programs to forecast 'The Markct', weather, ...

Take an active role in your child's development.

Parenting. The most important and rewarding endeavor you'll ever undertake. Gaze into your child's eyes. They're capturing all the wonders of the world around him, and looking to you for guidance.

Now you can gain a unique insight into your child's world with Childpace™ — an amazing new Child Development Program for ages 3 to 60 months.

Share the precious firsts.

When will your baby dazzle you with his first spontaneous smile? Stand alone? Take that first wobbly step?

The first five years are filled with continual growth and change. And questions. So even if your child's a toddler, you're still looking for answers. When will he start dressing himself? When should those random scribbles turn into distinctive shapes?

Compare apples-to-apples.

Childpace lets you evaluate your child's dexterity, language and social skills in the privacy of your own home. You enter information into Childpace, then he attempts tasks that are appropriate for his age group.

Childpace assesses his skill level based on extensive research, not the biased opinions of friends or relatives. Childpace uses your child's chronological (actual) age.

Grow with your child.

As your child grows, the tasks change to match his newly acquired skills. So Childpace is just as valuable for a 48-month old child as for an infant. Childpace can even evaluate up to 16 different children, and keep permanent records on each of them. Snapshots record your child's physical growth, but Childpace documents his or her actual development.

Track your child's progress, and help him develop specific skills. Childpace also contains warning signals to alert you to potential developmental problems at an early age, before they hold your child back. An ounce of prevention pays off.

Childpace. A fascinating glimpse into the world of child development. And more importantly, into *your child's* world.

Look for Childpace at your local computer hardware or software store. If unable to find it, send $39.95 to Computerose, Inc. Please allow two weeks for processing. 30 day money back guarantee.

$39.95 suggested retail price

Childpace is available for the Commodore 64®, IBM PC®, IBM PC Jr.,® Atari 800®, Apple II® and Radio Shack Color Computer®*

*Each is a registered trademark of the respective manufacturer.

Computerose™
We're programming for life.

2012 East Randol Mill Road Suite 223
Arlington, TX 76011 (817) 277-9153